NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGN OF AN 8 x 8 NON-BLOCKING CROSSPOINT SWITCH IN GaAs TWO-PHASE DYNAMIC FET LOGIC

by

David R. Bates

December, 1997

Thesis Advisor: Douglas J. Fouts
Second Reader: Sherif Michael

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1997	Master's Thesis

4. TITLE AND SUBTITLE
DESIGN OF AN 8 x 8 NON-BLOCKING CROSSPOINT SWITCH IN GaAs TWO-PHASE DYNAMIC FET LOGIC

5. FUNDING NUMBERS

6. AUTHOR(S)
Bates, David R.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Computer resources on military and telecommunications satellites are being over-tasked more than ever before, and the increasing shift to onboard signal processing will only compound the problem in the future. Space-based multiprocessor computer systems linked by high speed interconnect networks offer one possible solution to this ever expanding problem. Gallium arsenide (GaAs) integrated circuits using metal-semiconductor field effect transistors (MESFETs) offer very high speed operation, reduced power consumption, and inherent radiation tolerance, which make them ideally suited to the harsh space environment.

The design, simulation and layout of an 8 x 8 non-blocking crosspoint switch implemented in GaAs two-phase dynamic FET logic (TDFL) is presented in this thesis. The design of the TDFL crosspoint switch given here is a modification of a crosspoint switch design that uses GaAs direct-coupled FET logic (DCFL). Design specifics of working with GaAs are presented first, followed by detailed descriptions of the DCFL and TDFL crosspoint switches, and finally, an analysis of the advantages and disadvantages of dynamic logic over static logic is presented.

The TDFL crosspoint switch presented here could easily be modified to serve as a one gigabit per second serial interconnect for future space-based multiprocessor computer systems.

14. SUBJECT TERMS	15. NUMBER OF PAGES
GaAs, gallium arsenide, crosspoint switch, TDFL, two-phase dynamic FET logic	93
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298(Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

DESIGN OF AN 8 x 8 NON-BLOCKING CROSSPOINT SWITCH IN GaAs TWO-PHASE DYNAMIC FET LOGIC

David R. Bates
Lieutenant, United States Navy
B.S., United States Naval Academy, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1997

ABSTRACT

Computer resources on military and telecommunications satellites are being over-tasked more than ever before, and the increasing shift to onboard signal processing will only compound the problem in the future. Space-based multiprocessor computer systems linked by high speed interconnect networks offer one possible solution to this ever expanding problem. Gallium arsenide (GaAs) integrated circuits using metal-semiconductor field effect transistors (MESFETs) offer very high speed operation, low power consumption, and inherent radiation tolerance, which make them ideally suited to the harsh space environment.

The design, simulation and layout of an 8 x 8 non-blocking crosspoint switch implemented in GaAs two-phase dynamic FET logic (TDFL) is presented in this thesis. The design of the TDFL crosspoint switch given here is a modification of a crosspoint switch design that uses GaAs direct-coupled FET logic (DCFL). Design specifics of working with GaAs are presented first, followed by detailed descriptions of the DCFL and TDFL crosspoint switches, and finally, an analysis of the advantages and disadvantages of dynamic logic over static logic is presented.

The TDFL crosspoint switch presented here could easily be modified to serve as a one gigabit per second serial interconnect for future space-based multiprocessor computer systems.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Current integrated circuit (IC) technology is dominated by designs using the various silicon (Si) processes, but advances in gallium arsenide (GaAs) technologies and processes now give digital logic designers an alternative. Large scale integration (LSI) has been possible for GaAs ICs for many years. Recently, very large scale integration (VLSI) densities have been obtained with high yields on commercial parts and gate counts in excess of 100,000 gates. [1]

GaAs devices and circuits using field effect transistors (FETs) have numerous advantages over their Si counterparts, resulting primarily from the basic material properties of GaAs. Electron mobility in GaAs is more than five times greater than that in Si, therefore GaAs circuits switch faster and operate over a wider range of frequencies than Si circuits. GaAs ICs use a semi-insulating substrate that reduces parasitic and interconnect capacitance, as well as helping to isolate adjacent devices. GaAs is inherently resistant to ionizing radiation, as well as high heat, and is therefore an ideal choice for military and space applications.

Much of today's land-based communications infrastructure will be moving to satellite-based systems in the very near future. This shift to space systems will require that powerful digital processing capabilities make the shift to satellites as well. Multiprocessor computer systems with high speed interconnects will play vital roles in these emerging technologies. GaAs devices, with their high speed and radiation tolerance, are well positioned to fill this particular niche in the satellite communication and computing arena.

In order to obtain maximum performance benefit in a multiprocessor computer system, all processors must be able to communicate with each other in an efficient, high-speed manner. The non-blocking crosspoint switch is a simple, high-speed method for implementing a serial interconnect network between processors. The crosspoint switch is non-blocking because there is a dedicated communication path between every possible combination of input to output.

This thesis covers the design, simulation and layout of an 8 x 8 non-blocking crosspoint switch that could be used as the interconnect of a space-based multiprocessor computer system. The crosspoint switch is designed in GaAs two-phase dynamic FET logic (TDFL), and is functionally equivalent to a crosspoint switch designed in previous research using GaAs direct-coupled FET logic (DCFL). The crosspoint switch is being implemented in both DCFL and TDFL in order to examine the advantages and disadvantages peculiar to each logic topology. Background information on GaAs material properties and device characteristics appear in Chapter II. Chapter III covers a description of the DCFL crosspoint switch and Chapter IV covers the detailed design, simulation and layout of the TDFL crosspoint switch. Finally, Chapter V gives analysis and conclusions of the design effort and lists some recommendations for follow-on research.

II. BACKGROUND

A. REVIEW OF GaAs IC DEVELOPMENT

GaAs first saw use as a semiconductor material in the 1960s, with the first field effect transistors being developed in 1970. [2] Since that time, GaAs ICs have occupied a unique niche market where high speed operation and/or operation in harsh environments is of great importance to designers.

Early attempts at useful GaAs circuits were limited by low wafer yields and relatively low levels of integration. These limitations aside, GaAs quickly become the material of choice for high speed microwave devices such as amplifiers, oscillators, mixers, attenuators, limiters and switches. GaAs also has very desirable optical qualities and found heavy use in the early development of light emitting diodes, lasers and photo-detectors. GaAs use in this area is beginning to give way to more exotic solutions such as Indium Phosphide (InP) and InGaAsP because of their better wavelength properties in fiber optic systems. [3]

The first high-speed digital circuit applications were developed in the mid 1970's using small scale integration (SSI) to produce frequency divider circuits. Since that time, integration levels have increased steadily to the point where VLSI densities are becoming commonplace. Advances in the reliable production of GaAs wafers is one of the reasons for the increase in integration density and yield that the industry is enjoying today. IC fabrication and lithography techniques have also been improved.

Although GaAs yields and integration are increasing rapidly, they are still very far behind the levels common in the Si complementary metal oxide semiconductor (CMOS) processes in use today. Many of the lessons learned in the growth of high quality Si wafers and Si IC fabrication have been ported to the GaAs industry and resulted in much improved wafers and ICs.

The bulk of GaAs digital design today uses n-channel metal semiconductor FETs (MESFETs) as the switching devices. While p-channel MESFETs are possible, the hole mobility in GaAs is relatively slow, so there is no advantage in producing them. Current research in the field is exploring even faster and lower power switching transistors such as high electron mobility transistors (HEMTs) to further increase the performance gap over comparable Si processes.

B. MATERIAL PROPERTIES OF GaAs

This section briefly summarizes the basic material properties of GaAs that give rise to the performance characteristics observed in GaAs ICs. The qualities of GaAs circuits such as high speed, radiation tolerance, heat dissipation, and even the use of MESFETs instead of metal oxide semiconductor FETs (MOSFETs) are all the result of these basic material properties.

1. High Electron Mobility

GaAs circuits are much faster than their Si counterparts because of a much greater carrier mobility in the GaAs FETs. The carrier mobility is a measure of how easy it is for charge carriers to drift within a given substrate material. Carrier mobility is obtained by dividing the carrier drift velocity (cm/s) by the applied electric field (V/cm) and has the

dimensions $cm^2/V \cdot s$. [4] The carrier mobility is drastically different for electrons and holes in GaAs, with the electron mobility eclipsing the hole mobility by a factor of 20. This is the primary reason that GaAs devices are predominantly n-channel and why there has been so little success with complementary GaAs logic.

In comparing GaAs carrier mobilities with those of Si, the strength of the applied electric field is very important, as shown in Figure 1.

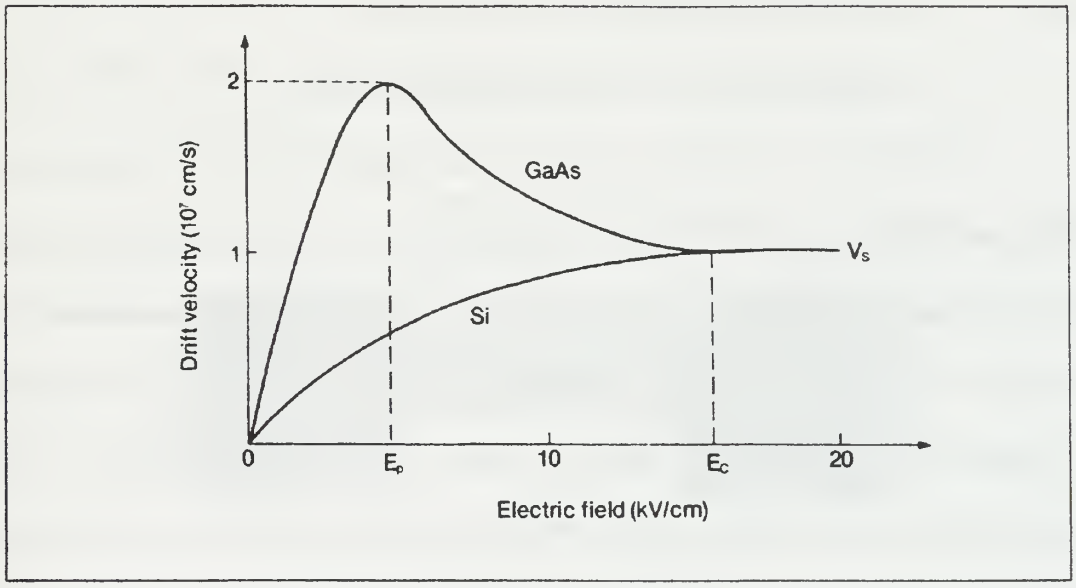


Figure 1. Equilibrium Electron Drift Velocity in GaAs and Si. From Ref. [5]

In high electric fields (10-100 kV/cm), the saturated drift velocity of electrons in GaAs is 8×10^6 cm/s, which is only slightly higher than the 6.5×10^6 cm/s figure for Si MOSFETs. Where GaAs really excels is in the lower electric fields characteristic of high-speed, low-power ICs. The peak electron velocity of $1.7\text{-}10^7$ cm/s for GaAs occurs at

E=3.5 kV/cm and gives a peak electron mobility of 5000 $cm^2/V \cdot s$ which is over 8 times the value of 800 $cm^2/ V \cdot s$ for Si with equal donor concentrations of $N_D = 10^{17}/cm^{-3}$.

In contrast, the hole mobility in GaAs is actually lower than that for equally doped p-type Si. The low electric field hole mobility for Si is about 300 $cm^2/V \cdot s$, but only 250 $cm^2/ V \cdot s$ for GaAs. Although the hole mobilities for Si and GaAs are very comparable, the narrower gap between n-type and p-type Si carrier mobilities has made complementary logic commercially attractive for Si. The much larger gap between n-type and p-type GaAs has made complementary logic commercially unviable.

2. High Resistivity or Semi-Insulating Substrate

Pure GaAs has a very high resistivity, in the range of 10^6-10^8 $\Omega \cdot cm$, and is often called semi-insulating GaAs. This semi-insulating state can also be obtained in less pure GaAs by properly doping the substrate with the correct ratio of donor and acceptor elements. One advantage of the high resistivity substrate is that interconnect capacitance is reduced over that found in Si. However, when connection lines are routed very close together, some of the advantage is lost. The more important advantage is that the semi-insulating substrate in GaAs helps to isolate adjacent devices, thereby reducing parasitic capacitance. Si has a much higher intrinsic carrier concentration and requires the use of reverse biased p-n junctions to isolate devices, which increases parasitic capacitance and decreases performance.

In addition to high resistivity, GaAs also has a very high surface state density. This, coupled with the fact that GaAs oxides are very unstable, prevents the production of GaAs (MOSFETs). Therefore, construction of GaAs FETs requires either the control

gate be laid directly over the conducting channel forming a Schottky-diode and called a metal semiconductor FET (MESFET), or control is via a p-n junction and is called a junction FET (JFET).

Si, on the other hand, has a very low surface state density and a very stable oxide in SiO_2 which allows very reliable MOSFETs to be constructed. Since the gate of a MOSFET is physically separated from the conducting channel, very little gate conduction occurs and a wide range of gate voltages are allowed. However, in a GaAs MESFET, the gate voltage is constrained by the forward conduction from gate to source. Typical cross sections of both a GaAs MESFET and a Si MOSFET are shown in Figure 2. [3]

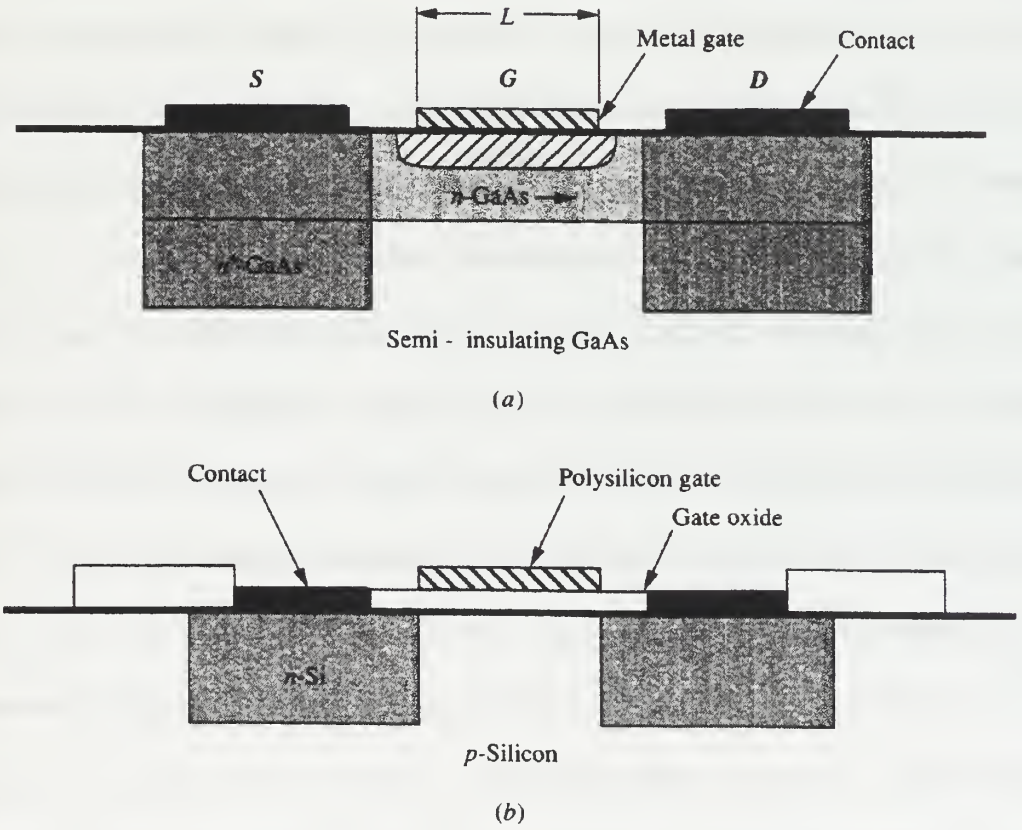


Figure 2. Typical Cross-Sections of (a) MESFET and (b) MOSFET. From Ref. [1]

3. Radiation Tolerance

The space environment can be very harsh on integrated circuits due to the much higher level of ionizing radiation compared with the terrestrial environment. GaAs devices have several characteristics that make them better than Si for surviving radiation exposure. The main problem with Si MOSFETs is that radiation causes charge to build up in the oxide layer under the transistor gate which has the effect of shifting the device threshold voltage. This is not a problem with GaAs MESFETs, as there is no oxide layer. Charge is absorbed by the surface layer of GaAs due to the high surface state density. By isolating the charge on the surface, the deep conducting channel of the MESFET is unaffected.

Over the long term, damage to the crystal structure of Si due to heavy ion damage can be quite significant and while some of this type of damage affects GaAs, the effects are much less prevalent due to the chemical bonds in the compound-semiconductor. Shorter-term effects such as dose-rate effects are also less prevalent in GaAs thanks to the high resistivity substrate which restricts spurious current flow.

Some single-event effects such as single-event latchup (SEL) are also less of a problem in GaAs ICs. Since there are no well-substrate junctions in GaAs ICs, GaAs circuits are immune to SEL. All of these characteristics combine to make GaAs a much better semiconductor choice for high-speed digital space-based applications. [6]

C. GaAs MESFETs

As stated previously, intrinsic GaAs has a very high resistivity and is considered a semi-insulator. To create useful semiconductors, dopants must be added to the GaAs in order to increase the numbers of free-charge carriers. When the majority carriers are

electrons, negative charge is transported and the material is referred to as n-type. When the majority carriers are holes, positive charge is transported and the material is called p-type. To create n-type GaAs, dopants that create surplus electrons like Se, Te, S, Si or Sn are implanted. To create p-type GaAs, dopants that capture surplus electrons such as Zn, Be, Mg, Cd and C are implanted.

The large performance difference between n-type and p-type GaAs devices described previously is the primary reason that complementary logic devices are not used as widely as they are with Si. P-type GaAs devices are rarely used in industry and all of the devices used for the design of this thesis project use n-type GaAs.

D. ENHANCEMENT AND DEPLETION MODE MESFETs

The operating characteristics of MESFETs are determined by the energy and dose of the device's channel implant. A MESFET with a light channel implant is called an enhancement-mode device or EFET and has a positive threshold voltage. If the channel implant is moderate, the threshold voltage becomes negative and the device is called depletion-mode or DFET. A more rare device exists called an MFET that has a heavy channel implant and a large negative threshold voltage and acts like a strong depletion device. [8] The design presented in this thesis makes use of both EFETs and DFETs, with DFETs used as pull-ups to V_{DD} and pass-transistors and EFETs used to switch to ground and as reverse-biased diodes to store charge. Figure 3 shows the circuit symbols used to represent the different MESFETs.

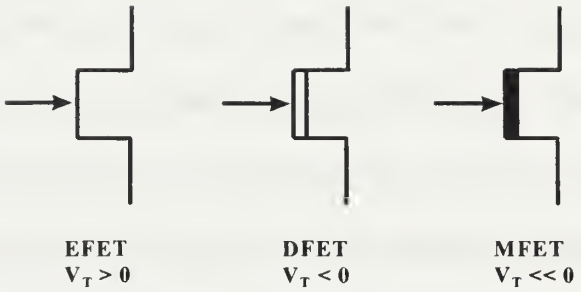


Figure 3. MESFET Circuit Symbols.

1. Enhancement-mode FETs

Enhancement-mode FETs have a positive threshold voltage, $V_T > 0$, and are considered normally "off". EFETs are produced by setting up the depletion region to extend through the entire conducting channel when the gate-to-source voltage is zero, thereby preventing any current to flow from drain to source. The "pinch-off" effect that blocks the channel is removed when a positive voltage pulls free electrons into the channel and allows current to flow between the drain and the source. The positive voltage applied to the gate is limited by the Schottky-barrier voltage at the gate-to-channel junction (approximately 0.7 V). If this voltage is exceeded, gate conduction occurs and a non-negligible amount of current flows from the gate into the channel, resulting in V_{GS} no longer controlling the drain to source current. Therefore, the logic levels allowed on the gate are severely restricted, which results in reduced noise margins. EFETs require very tight control of V_T in fabrication because slight variations can cause improper logic swings which reduce chip yield and ultimately increase unit costs.

2. Depletion-mode FETs

Depletion-mode FETs have a negative threshold voltage, $V_T < 0.0$ V, and are considered normally "on". The channel is moderately implanted such that there are enough free electrons under the gate to form a conducting path from drain to source with $V_{GS} = 0.0$ V. Threshold voltage for a typical DFET is approximately -0.85 V. When a negative voltage greater than V_T is applied to the gate, free electrons are repelled downward out of the channel, forming a depletion region that pinches off the channel near the drain and prevents current from flowing from drain to source. Thus, DFETs have the problem of needing a negative V_{GS}, to prevent drain current flow. This condition requires more circuit overhead in DFET-only logic circuits to make the input and output voltage polarities compatible. However, larger logic swings are allowed which improves the noise margins considerably for DFET-only logic circuits.

E. STATIC LOGIC CIRCUITS

1. Background

Static logic circuits do not rely on clocking mechanisms for proper operation. Logic levels and noise margins in static logic circuits are controlled by the length to width ratios of the active devices, hence static logic is synonymous with ratioed logic. In static logic circuits, input levels control current flows from power to ground in order to establish output levels. [3] There are numerous static logic families in use today but no real standard for circuit design. Static logic families currently in use include capacitor-diode FET logic (CDFL), Schottky-diode FET logic (SDFL), direct-coupled FET logic (DCFL), superbuffer FET logic (SBFL), and source-coupled FET logic (SCFL).

The existing design of the crosspoint switch was created with a combination of DCFL and SBFL static logic families. The new design uses dynamic logic to implement the same functionality. However, some use of static DCFL and SBFL logic gates was necessary. A brief description of DCFL and SBFL logic is given below.

2. DCFL Logic

DCFL is an enhancement-mode logic family and is the simplest and most widely used in the industry. A schematic of a simple DCFL inverter is shown in Figure 4.

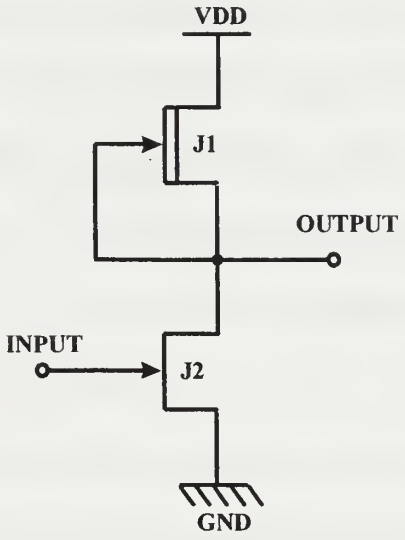


Figure 4. Schematic of DCFL Inverter.

The DFET J1 is a MESFET load that acts as a pull-up to V_{DD}, while the EFET J2 acts as a switch to control whether the output is at V_{DD} or ground, based upon the input voltage. As discussed previously, the input voltage is limited by the Schottky-barrier voltage, which has the benefit of increasing speed but unfortunately reduces the noise

margin of the circuit. For proper operation, the output low voltage must be very near zero to ensure the threshold voltage requirements of follow-on gates are met. The DCFL inverters used in this thesis came from a standard Vitesse HGaAs III library and have DFET dimensions of 1.6 μm long by 2.0 μm wide and EFET dimensions of 0.8 μm long by 16 μm wide. The disparity in dimensions is to provide for a more symmetric DC transfer curve.

DCFL logic gates are characterized by very few circuit elements per gate, which results in lower interconnect parasitic capacitance, higher gate density, lower power consumption, and higher speed than other GaAs logic families. [3] One area in which DCFL logic is at a disadvantage is in static power consumption. When the output of a DCFL inverter is low, there is a direct path from power to ground and considerable power is consumed. One of the primary goals of dynamic logic families is to break this direct current path from V_{DD} to ground.

3. SBFL Logic

Superbuffer FET logic is nearly identical to DCFL logic but uses a quasi-complementary output driver to increase the output drive capability. Figure 5 gives the schematic of an SBFL inverter.

Transistors J1 and J2 are a DCFL inverter, J3 is a wide EFET that acts as a pull-up to V_{DD}, and J4 is a switch to ground that is identical to J2. On a high output, the wide EFET is able to supply its full current to the load capacitance. Low-to-high transitions simply follow the inverter stage but high-to-low transitions can lead to a condition where

both J3 and J4 are momentarily conducting at the same time, which requires ample margin in the power bus design. [3]

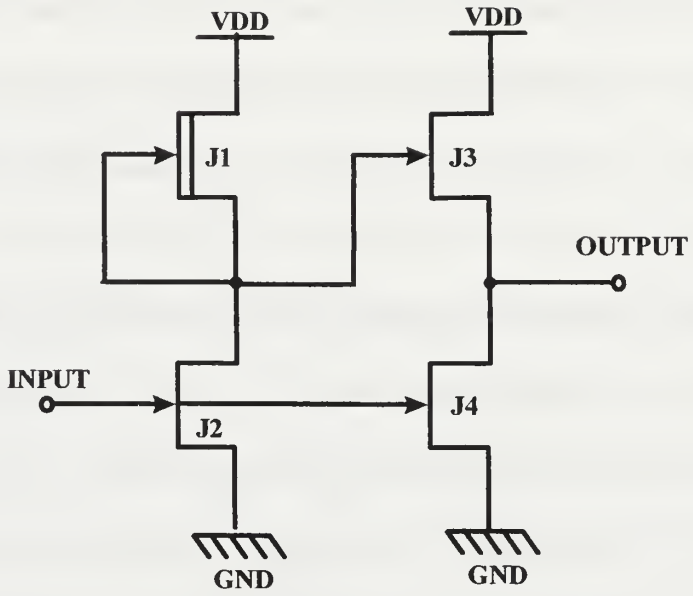


Figure 5. Schematic of SBFL Superbuffer.

An alternative version of the superbuffer uses a DFET in position J3, which increases performance on high-to-low transitions and also allows an output to fully reach V_{DD}. When used exclusively, superbuffers have better noise margins than DCFL inverters but when the two are mixed, noise margins migrate to the DCFL limits. Both types of superbuffers mentioned were used in the crosspoint switch design when output load capacitance dictated a large drive capability.

F. DYNAMIC LOGIC CIRCUITS

1. Background

Dynamic logic circuits perform their functions by storing and evaluating charge on isolated circuit nodes that are controlled by a clocking scheme. In contrast to static logic,

there is never a direct current path from power to ground so there is considerably less power consumption. Logic levels are not dependent on the length-to-width ratios of the devices, thus dynamic logic is often referred to as non-ratioed logic. Since the ratios are not used to determine logic levels, the device sizes can be chosen to reduce power and increase speed. Dynamic logic topologies are often able to take advantage of minimum-size devices as dictated by the design rules of a particular fabrication process.

While dynamic logic has been used extensively in Si CMOS designs, it is still a relatively rare solution in GaAs design. Dynamic logic families in use today include several single-phase domino logic topologies, trickle-transistor dynamic logic (TTDL), capacitively-coupled dynamic logic (CCDL), and two-phase dynamic FET logic (TDFL). The design of the crosspoint switch presented in this thesis uses the TDFL logic topology.

2. Theory of TDFL Operation

TDFL is a non-ratioed logic family that uses a single power supply and dissipates power only during transitions of its two non-overlapping clock signals. TDFL gates are compatible with the DCFL family and all logic functions are available. Figure 6 shows two TDFL inverters connected in series and Figure 7 shows an HSpice simulation of their operation. [7]

For this example, a power supply of 2.0 V was used and the non-overlapping clocks $\Phi 1$ and $\Phi 2$ toggle between -1.0 V and 0.5 V. The non-overlapping clocks enable TDFL circuits to accomplish their precharge-evaluate cycle that prevents the direct flow of current between power and ground.

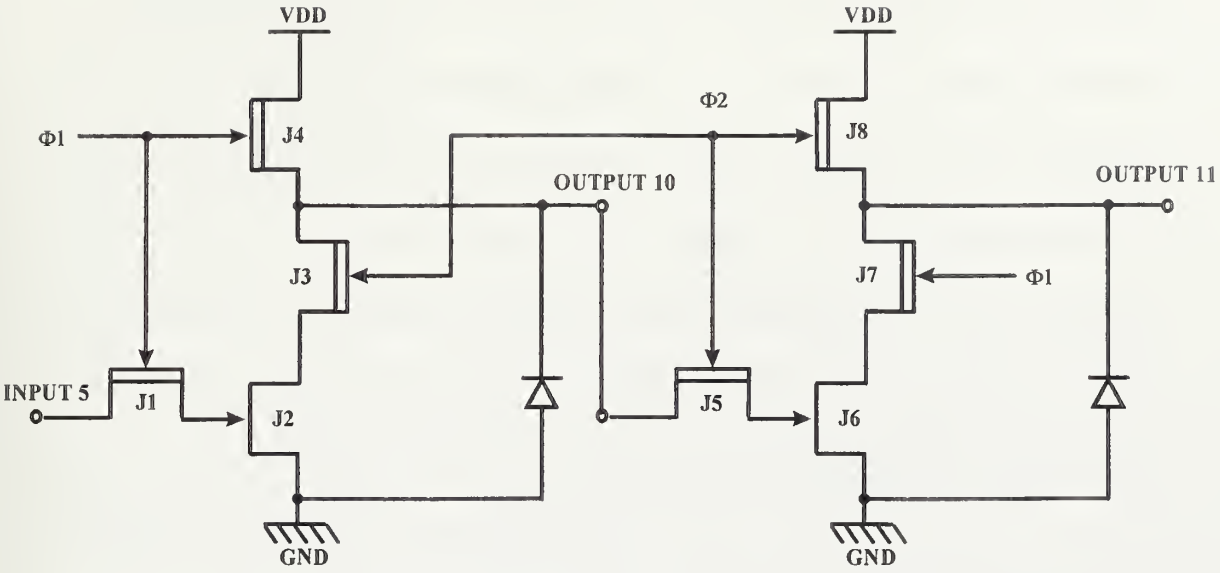


Figure 6. Schematic of Two TDFL Inverters in Series.

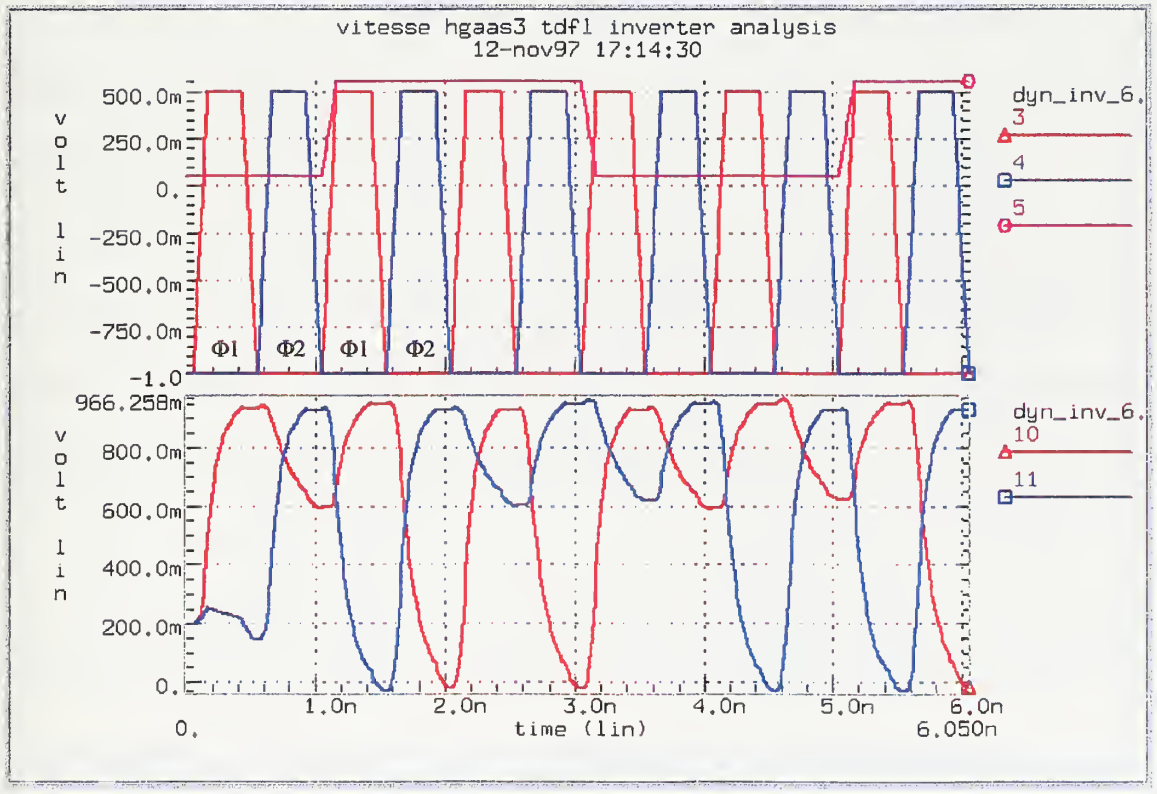


Figure 7. Simulated Operation of Inverters with V_{DD} = 2.0 V.

When Φ1 is high, DFETs J1 and J4 are on and the input voltage moves to the gate of J2 while the output node is precharged high. Next, Φ1 goes low, trapping the charges in the now isolated nodes. When Φ2 goes high, J3 and J5 turn on which evaluates the logic value of the first stage and allows it to propagate to the gate of EFET J6 in the second inverter. In this example, the input was low which kept EFET J2 off. Therefore, on the evaluate stage the output which was precharged high was able to float and remain high. An EFET configured as a reverse-biased diode is used to add capacitance to the output node in order to increase the gate fan-out.

It is important to note that the output of a TDFL gate is only valid during the evaluate clock cycle and will always be high after the precharge clock cycle. This means that for TDFL gates to be connected to each other, the clocks driving the precharge and evaluate cycles must be alternated as shown in Figure 6. In other words, if the first gate evaluates on Φ2, then the next gate must precharge on Φ2 and evaluate on Φ1. This alternation induces a one-half clock period delay due to the precharge taking one-half clock cycle, and can be seen in Figure 7 where output11 lags output10 by one-half clock cycle. This feature of TDFL logic does slow the output slightly but makes this logic family an excellent choice for implementing pipelined designs. [1]

3. Design Considerations

Much of the correct operation of TDFL circuits is dependent upon the storage of charge on isolated circuit nodes. This fact makes the nodal capacitances of the transistors very important. The use of minimum-sized transistors to form the gates will reduce power consumption and increase speed. However, the small devices might not have enough

nodal capacitance to store enough charge for circuits with moderate to large fan-outs. Figure 8 is an HSpice simulation that shows the effect of large fan-outs on the output of a TDFL inverter.

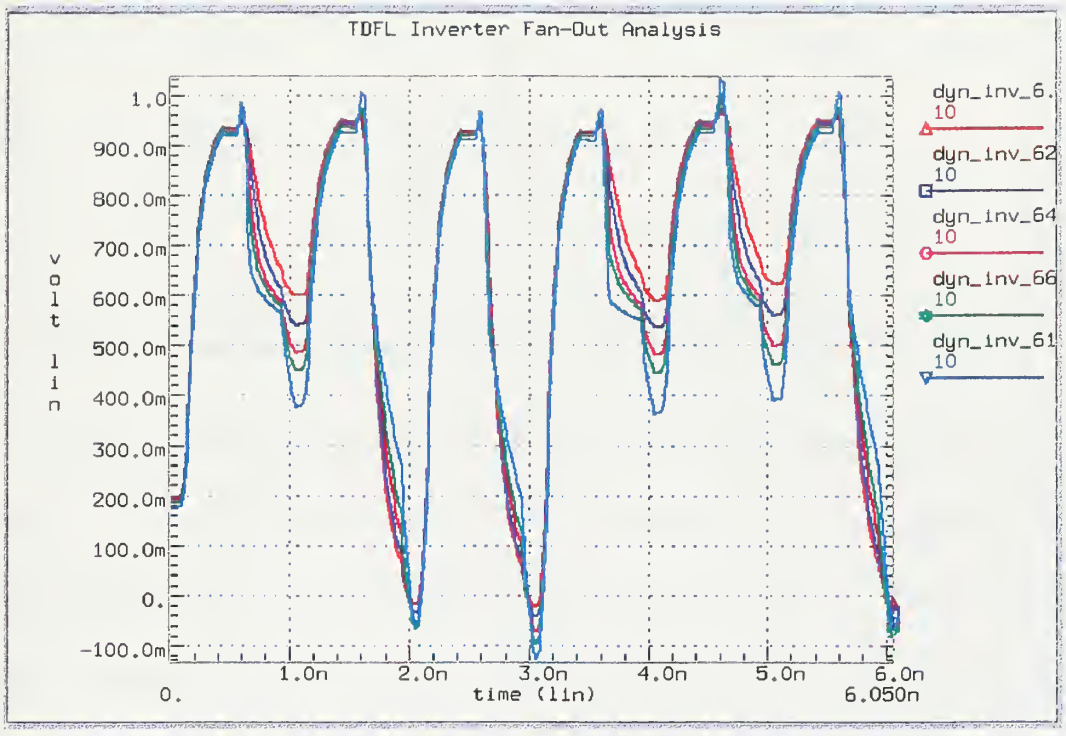


Figure 8. Simulated Effect of High Fan-Out on TDFL Inverter Output Levels.

The different plots show the effect of a fan-out of 1, 2, 4, 6 and 12 follow-on inverters. Since the output high state is floating, increased fan-out causes the output voltage to drop significantly, which adversely affects the noise margins.

The size of the switching EFET also plays an important role in TDFL design. A smaller EFET will put less drain on an output high signal but will provide a restricted pull-down on an output low signal.

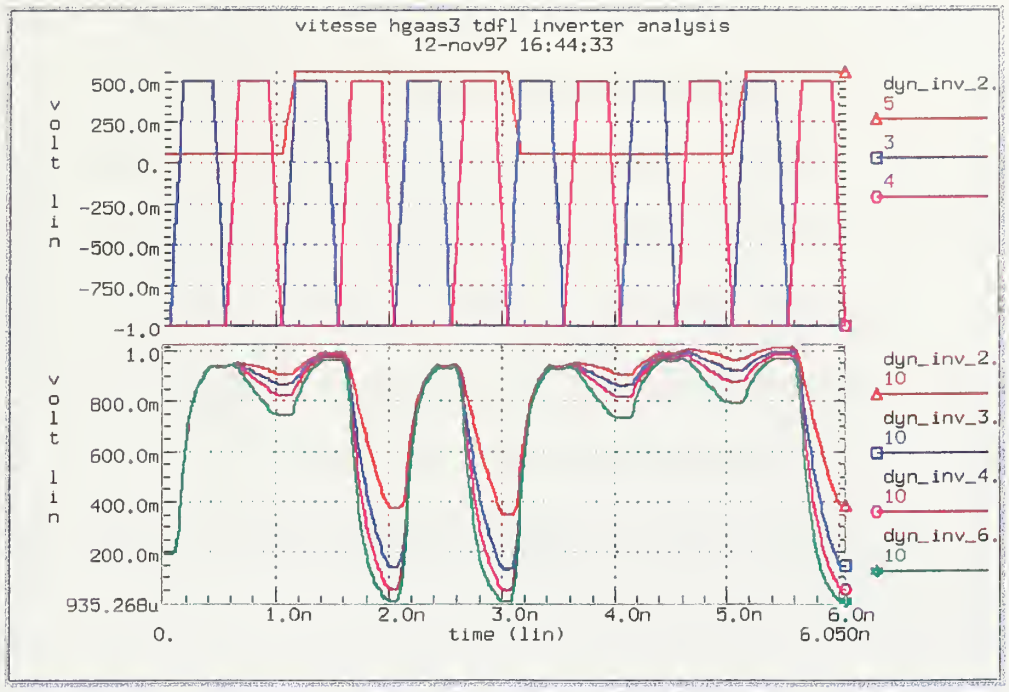


Figure 9. Simulation of TDFL Inverter with 2, 3, 4, and 6μm Wide EFETs.

Figure 9 shows the effects of varying the width of the switching EFET for 2, 3, 4 and 6 μm. The highest plot in the lower window shows that a 2 μm wide EFET gives a solid floating-high output but can only pull a low output down to 0.4 V. The 6 μm width is the lowest plot and provides a much better compromise between logic low and logic high values. Widths above 6 μm provide too much capacitance and give unacceptable low values for a logic high.

The reverse-biased diode attached to the output node further complicates the capacitance issues. The TDFL designer must consider all of these issues when deciding on the proper device sizes to use in a particular design.

Another major issue the TDFL designer must consider is how to get the clock signals to all of the TDFL components. The designer must decide if the non-overlapping clocks will be brought on-chip or if they will be produced on-chip from a single-phase

clock. Since each device requires both clock signals, routing can become quite messy. For large designs, care must be taken to prevent excessive clock skew. TDFL logic is somewhat tolerant of small to moderate clock skew but large clock skews can cause large problems, especially if the clocks possess some small amount of overlap. [1]

Overall, designing in dynamic logic is much more complicated than static logic. However, the designer has a great deal of freedom in the design tradeoffs and the end results are circuits that exhibit very high speed with very low power consumption.

III. 8 x 8 NON-BLOCKING CROSSPOINT SWITCH

The design goal of this thesis was to produce an 8 x 8 non-blocking crosspoint switch in GaAs TDFL with the same functionality as a previously designed DCFL crosspoint switch. These switch designs are part of a building block process to test the feasibility of designing a much larger crosspoint switch that could be used in space-based multiprocessor computer networks. By comparing results between the static and dynamic versions of the switch, design trade studies can be made early in the process, before a large scale switch project is undertaken.

A. FUNCTIONAL DESCRIPTION

Crosspoint switches were originally designed for the switching of analog signals. However, their use has become widespread for the switching of digital signals as well. Crosspoint switches are in a class referred to as space-division switches because the various signal paths are physically separated from each other. Figure 10 shows a behavioral representation of an 8 x 8 crosspoint switch.

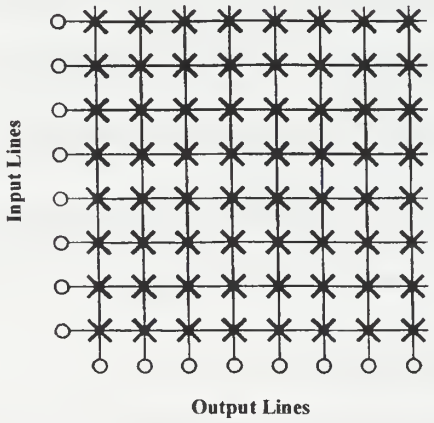


Figure 10. 8 x 8 Crosspoint Switch. From Ref. [9]

The switch is made up of 8 input lines, 8 output lines and 64 crosspoints. Connecting any output with any input is as simple as enabling the crosspoint where the lines intersect. In reality, the behavior of the switch is realized with eight 8-to-1 multiplexers. Each connection that is made establishes a physical path through the switch that is dedicated solely to the transfer of data from that particular input to the selected output. The fact that there is a crosspoint for every combination of input to output connection makes this type of switch very simple to control. Having every combination of switch also makes the crosspoint non-blocking, as there will always be a path from any input to any output. There are several disadvantages to this type of switch which might pose problems in future designs. The number of crosspoints required grows with the square of the number of input/output lines. This can be a very large number of crosspoints in a large switch. If a crosspoint fails, the connection at that intersection point is lost and cannot be regained. Finally, the crosspoint switch design is very inefficient because when all lines are connected, only a small percentage of the crosspoints are actually used.

B. DCFL CROSSPOINT SWITCH DESIGN

As stated previously, the DCFL crosspoint switch has already been designed and fabricated and is awaiting testing. The DCFL design was completed prior to the start of this thesis. Thus, none of the work presented here contributed to that design effort. The high-level design of the DCFL crosspoint switch is presented to illustrate the functionality as well as the changes necessary to port the design to the TDFL topology.

1. 8-to-1 Multiplexers

Figure 11 shows a schematic of one of the 8-to-1 multiplexers used in the design.

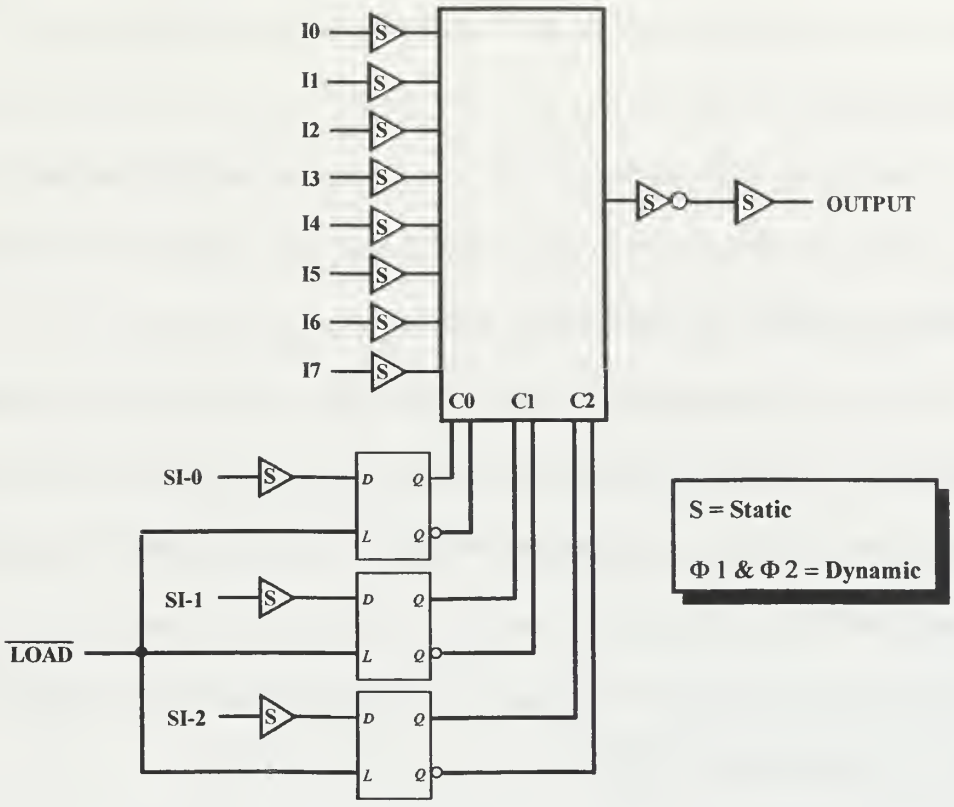


Figure 11. DCFL 8-to-1 Multiplexer High-Level Schematic.

At the heart of the DCFL crosspoint design there are eight 8-to-1 multiplexers. The eight multiplexers are identical to each other, with the exception of the LOAD signal, which is a unique signal for each multiplexer that comes from a separate output select circuit that will be described later. Each of the multiplexers has a single output line that corresponds to one of the eight switch outputs. The eight input lines to the multiplexer correspond to the eight switch inputs and are identically routed to each of the eight multiplexers. Each of the multiplexers can be thought of as one of the columns of the behavioral model in Figure 10.

Inputs are brought onto the chip via non-inverting, F100K, ECL to DCFL input receivers. Input receivers provide electrostatic discharge (ESD) protection and accept

input voltage ranges from 0.2 V to 1.2 V and output a level of 0.05 V to 0.65 V to the DCFL chip core.

The outputs of the multiplexers are sent through a superbuffer to get the correct logic level with sufficient drive current and then through a DCFL to F100K ECL non-inverting output driver with ESD protection.

The crosspoints inside the multiplexer are controlled by the three select-input lines and the LOAD lines that are routed to the inputs of the 3-bit D Latches. The connection is made when the 3 select-output lines assert the LOAD line for a particular multiplexer, at the same time that the input lines are asserted. If the LOAD line is asserted, the input line is latched and the connection from input to output is held until a new select-input line is loaded into that particular multiplexer.

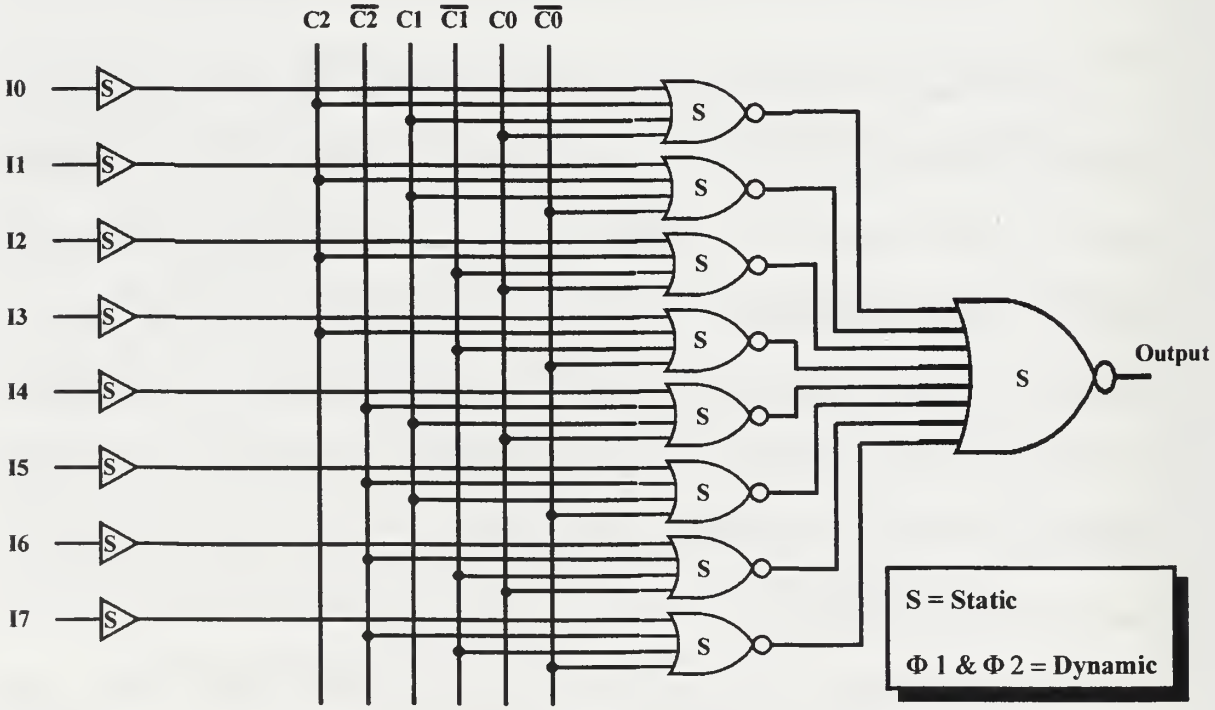


Figure 12. Schematic of DCFL Multiplexer Internal Circuitry.

The internal circuits of each multiplexer are shown in Figure 12. Eight 4-input NOR gates serve as the actual crosspoints and only one of them can be active at a time inside a particular multiplexer.

2. D Latches

As shown previously in Figure 11, the input line number is held in a 3-bit D Latch. The value of the input line is only latched if the LOAD signal is asserted low for a particular output line number. The D Latches are identical and are formed by the simple cross-coupling of 2-input NOR gates, as shown in Figure 13.

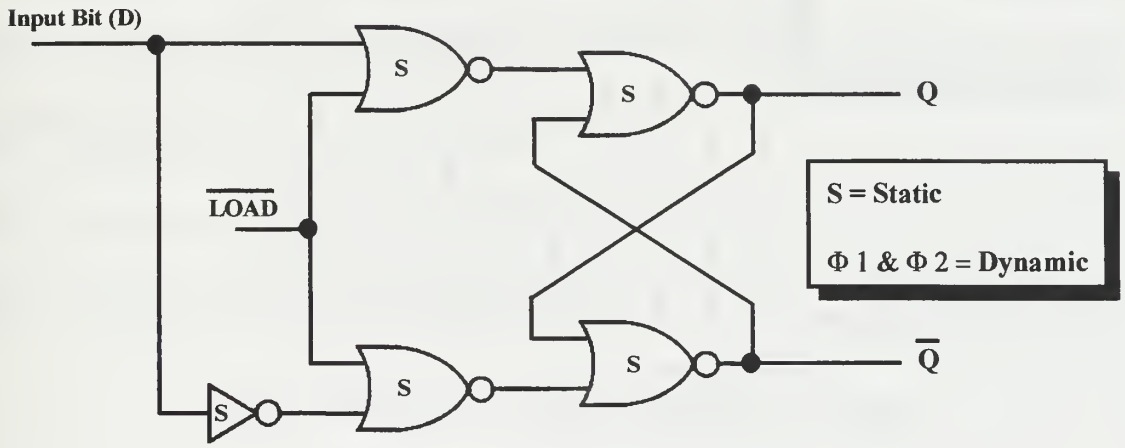


Figure 13. Schematic of Single DCFL D Latch.

The 3-bit latch will continue to hold an input line number until a new load signal is sent to latch a new input line number.

3. Output Select Circuit

There is a single output select circuit that controls the LOAD signals used to select an output multiplexer by latching a given input line number. The simple circuit using inverters and NOR gates is shown in Figure 14.

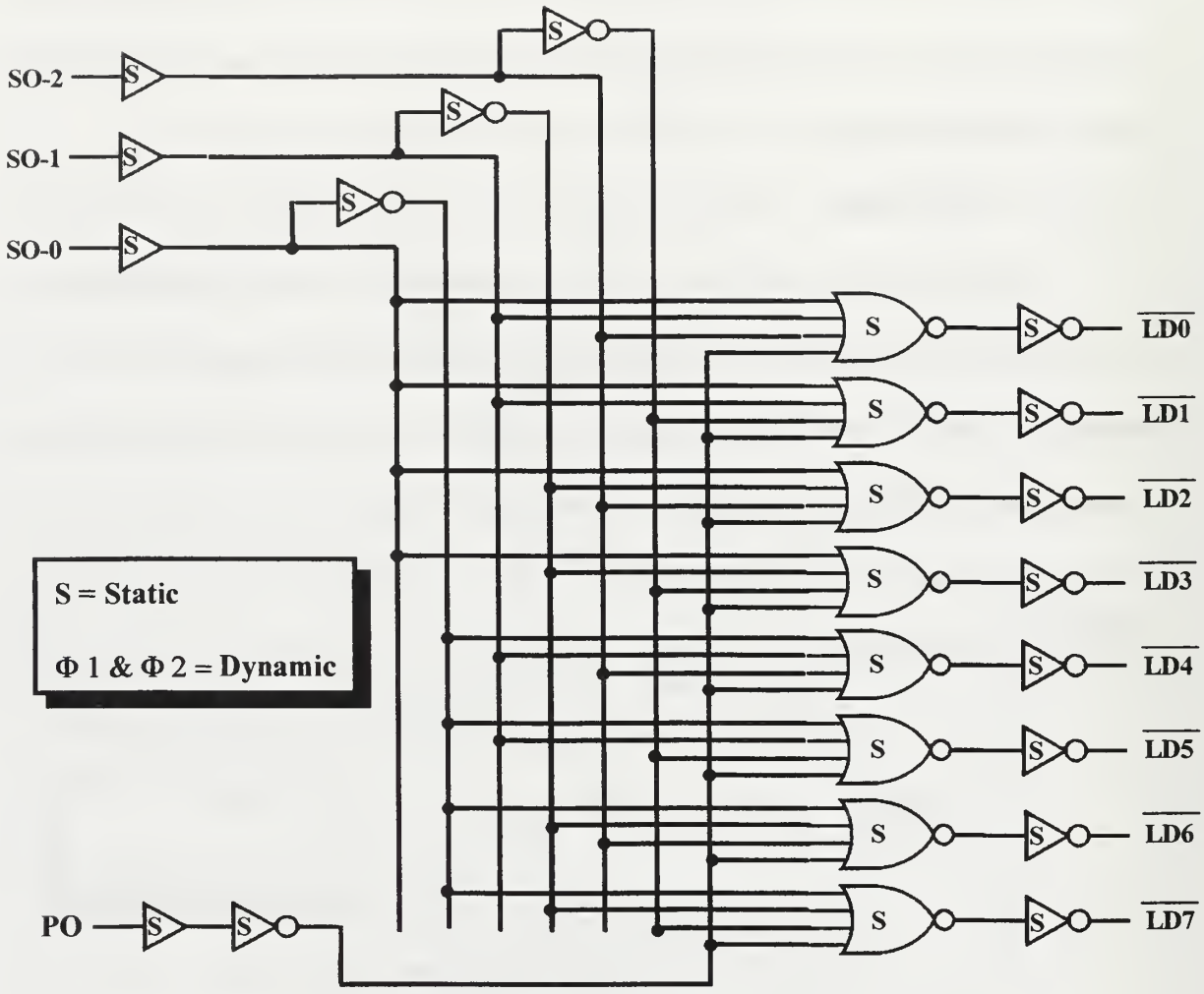


Figure 14. Schematic of DCFL Output Select Circuit.

The 3-bit value that defines an output line number forces a single LOAD line to be asserted low, which forces a specific latch to route an input to a single output. There is also a single input called Program Output (PO) which controls when a LOAD signal can be asserted. The Select Output (SO) lines and the PO line are all buffered by input receivers. The PO line is sent through an additional superbuffer to make it asserted low and to give it a fan-out capability of 8.

4. Operation

The crosspoint switch is designed to be set to a particular configuration and then left for some period of time to allow data transfers to take place. While it is easy to reconfigure the input/output connections, speed of reconfiguration is not a driving factor in the design. The reliability and very high speed operation of the connections themselves are more important to the overall design.

Once the operation of the DCFL crosspoint switch was fully understood, work began on recreating the design in the TDFL topology. The initial design goal was to exactly replicate the functionality of the DCFL crosspoint switch using TDFL structures. The step by step process used to design the TDFL crosspoint switch is presented in the following chapter.

IV. TDFL CROSSPOINT SWITCH DESIGN

The first step in designing the TDFL crosspoint switch was becoming familiar with the software tools that would be necessary to complete the project. HSpice was used to perform the circuit simulations and Magic was used to perform the actual layout. At some time in the future, the TDFL crosspoint switch will be submitted through MOSIS for fabrication by Vitesse Semiconductor Corporation of Camarillo, California.

A. DESIGN OF BASIC TDFL GATES

The actual design process started by building up a library of TDFL logic gates that would be used to construct the higher level structures required in the crosspoint switch. The gates that needed to be developed were an inverter and two, three and eight-input NOR gates. While there are only four types of basic gates needed, a great deal of simulation was required to get fully functional circuits. Nodal capacitance, fan-out, and change sharing issues forced the gate designs to be changed many times in order to arrive at an optimal solution.

1. TDFL Inverter

The TDFL inverter was the first gate designed and was used to explore how changing device sizes and fan-outs affected gate outputs. The effects of fan-out and varying switching EFET sizes were shown in Figures 8 and 9. The schematic for the TDFL inverter used in the design is shown in Figure 15.

All of the transistors used are minimum-size with the exception of the EFETs J4 and J5. The minimum-size DFETs measure 0.8 μm long by 2.0 μm wide. Using

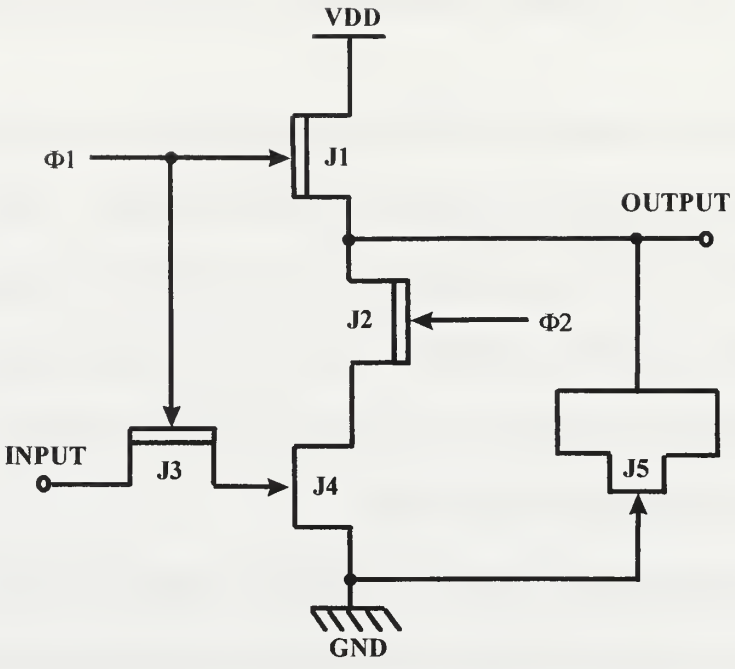


Figure 15. Schematic of TDFL Inverter Design.

minimum-size devices has the benefits of reducing input capacitance, reducing power consumption and reducing layout area. However, the smaller devices take longer to precharge the output nodes high. The speed of the precharge is not as important as the level of the precharge, so this disadvantage is disregarded.

The width of EFET J4 was optimized to provide a solid low near 0.0 V and a high that would not sink below about 0.6 V for a fan-out of 2. The final dimensions of J4 came out to be 0.8 μm long by 6.0 μm wide. The EFET J5 is configured as a reverse-biased diode to provide extra capacitance at the output node. The size of this EFET was determined by simulation trial and error and ended up with square dimensions of 9 μm by 9 μm. The reverse-biased diode configuration provides the greatest capacitance for the least amount of area.

2. TDFL NOR Gates

The schematic for the TDFL two-input NOR gate is given in Figure 16.

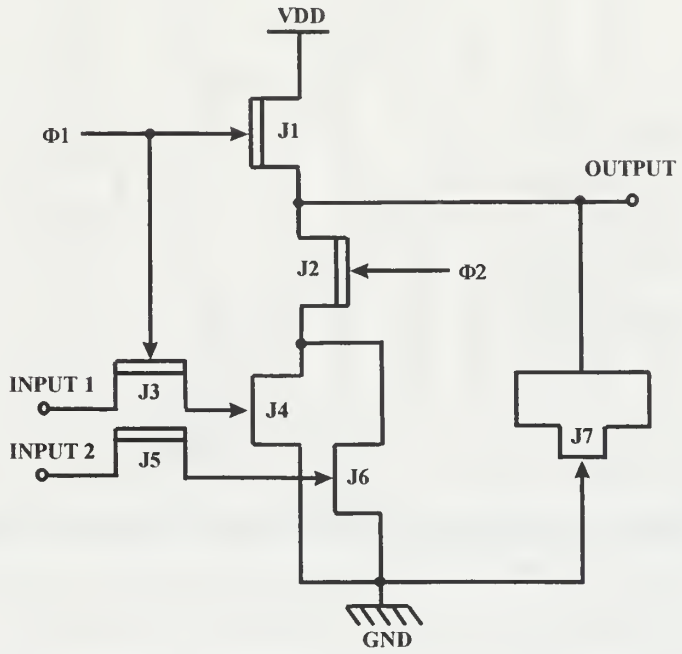


Figure 16. Schematic of TDFL 2-Input NOR Gate Design.

The two-input NOR gates in the design have a fan-out of only one, thus they could be made with all minimum-size devices without sacrificing noise margins. The output reverse-biased diode is slightly smaller than the one used in the inverter and measures 8 μm by 8 μm.

The gate that was used most often in the design was the four-input NOR gate, with a total of 72 instances. The schematic for a four-input NOR is given in Figure 17.

The four-input NOR gates appear in two different high-level structures and as a result required two different designs to achieve acceptable output levels. The NOR gates destined for the output select circuit were able to use minimum-sized transistors, with

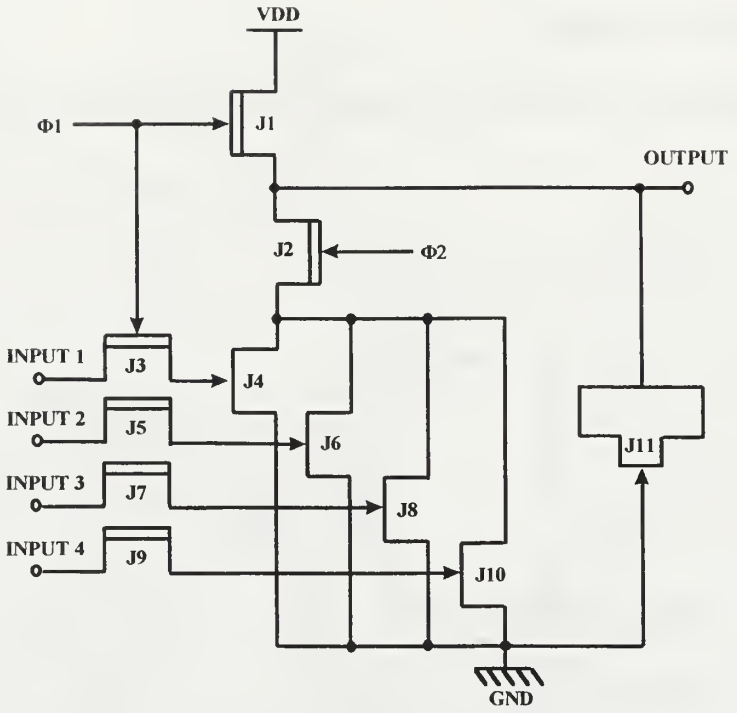


Figure 17. Schematic of TDFL 4-Input NOR Gate Design.

DFETs being 0.8 μm long by 2.0 μm wide and EFETs 0.8 μm long by 4.0 μm wide. The output diode was sized the same as the inverter at 9.0 μm x 9.0 μm. The NOR gates that were used inside the multiplexer circuits were driving a much larger capacitive load with the follow-on eight-input NOR. As a result, the switching EFETs had to be widened to 6.0 μm and the output diode had to be expanded to a much larger 12 μm x 12 μm. The four-input NOR gate is representative of the other NORs and a 1 GHz simulation is shown in Figure 18.

The very distinctive two-phase clock signals are labeled Φ1 and Φ2 along the bottom of the window. In this simulation, two of the input lines are tied low and the other two input lines are tied to the input waveforms shown in the plot. The output is the dark black waveform. It can easily be seen that the output precharges on the Φ1 phase and

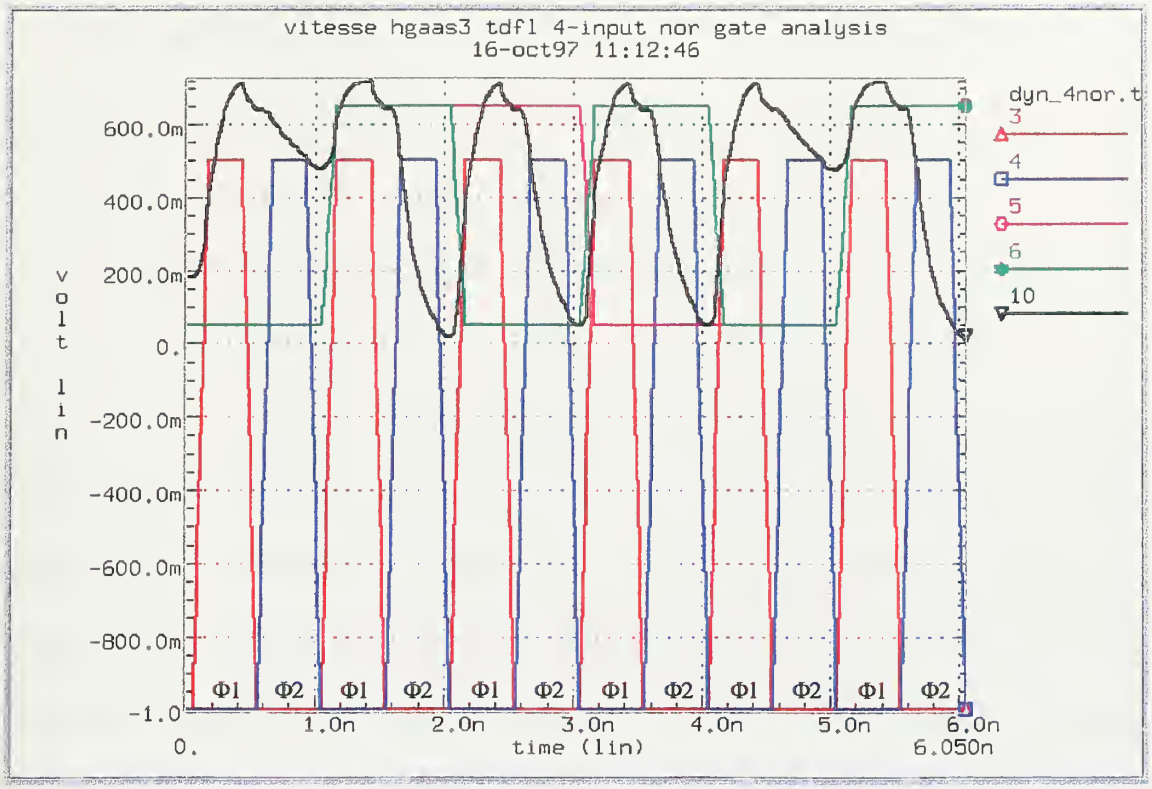


Figure 18. Simulation of TDFL 4-Input NOR at 1 GHz.

evaluates on the Φ2 phase. The first Φ1 precharges when both inputs are low. On the very next Φ2, the output evaluates high and the next Φ1 stage precharges when both inputs are high, thus the next Φ2 stage evaluates low. The output highs are about 0.5 V and the output lows are about 0.05 V, which give good TDFL noise margins.

The schematic for the eight-input NOR is a simple extension of the two and four-input gates and is omitted here. Before the eight input NOR was attempted, there was some concern that the parasitic capacitance of this gate would be too large, but those fears proved to be unfounded. The design was completed with minimum-size DFETs and EFETs with a 6.0 μm width. The output diode was sized at 12 μm by 12 μm to provide enough drive capability for the signal to reach the chip output drivers. Even in the worst

case scenario when only one input is high, forcing a single EFET to discharge all of the precharge to ground, the gate performed remarkably well. Noise margins for the eight input gate were very similar to those of the four input gates.

B. HIGHER LEVEL DESIGN AND SIMULATION

1. Basic Description

The design of the higher level structures used to create the TDFL crosspoint switch follow directly from the design of the DCFL switch. In most cases, the design required only to convert from DCFL gates to TDFL gates, with only a few minor modifications to account for the clocking of the TDFL gates. In several instances, DCFL inverters needed to be used in the TDFL design. When both a signal and its complement are needed at the same time, it is difficult to use TDFL inverters because they impose a one-half clock cycle delay. Also, when an output is needed for both phases of the clock, a DCFL inverter and a pass-gate can be used to latch the output of the TDFL gate.

In some instances, the input capacitance of a TDFL gate was not enough to clamp the output of the DCFL inverters to the desired range of 0.0 V to 0.65 V. These cases caused DC convergence problems in HSpice and required small capacitors in the simulation to ensure compatible voltage levels. The input receivers and output drivers are identical to those in the DCFL design.

2. Output Select Circuit

A schematic of the TDFL output select circuit is given in Figure 19. The only changes necessary to this circuit were the substitution of the TDFL four-input NOR gates and the addition of clocked pass-gates to latch the outputs of those NOR gates.

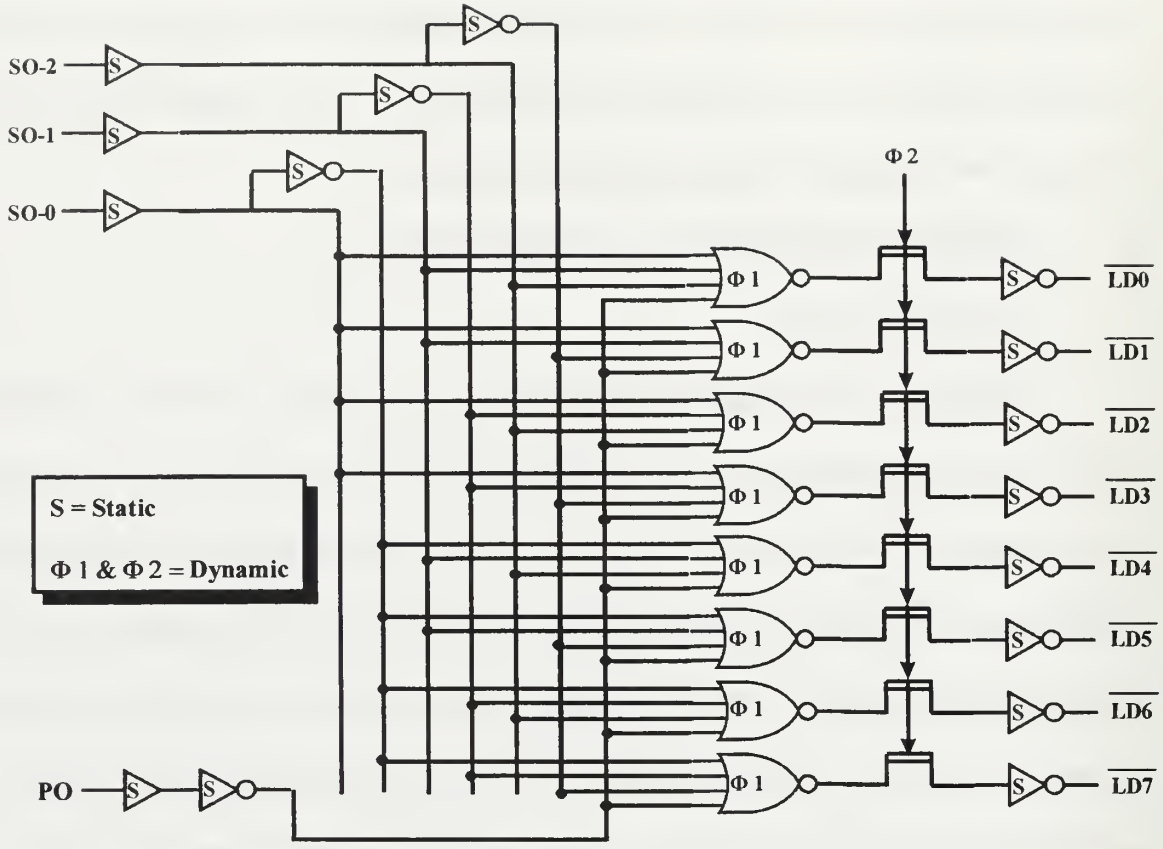


Figure 19. Schematic of TDFL Output Select Circuit.

The output select bits SO0-SO2 are required at the same time as their complements, therefore static inverters had to be used there. The fan-out of the Program Output bit is eight, thus a decision was made to use static gates since there is only one instance on the entire chip. The output LOAD signals required latching with static inverters because the D Latches in the multiplexers require the LOAD signal be available during both phases of the clock. A simulation of the output select circuit at 1 GHz is shown in Figure 20. The input waveform is PO prior to the input receiver and the output is an active low LOAD signal that is delayed one-half clock cycle from the input. The remaining seven LOAD signals all remain at the logic high level of approximately 0.7 V.

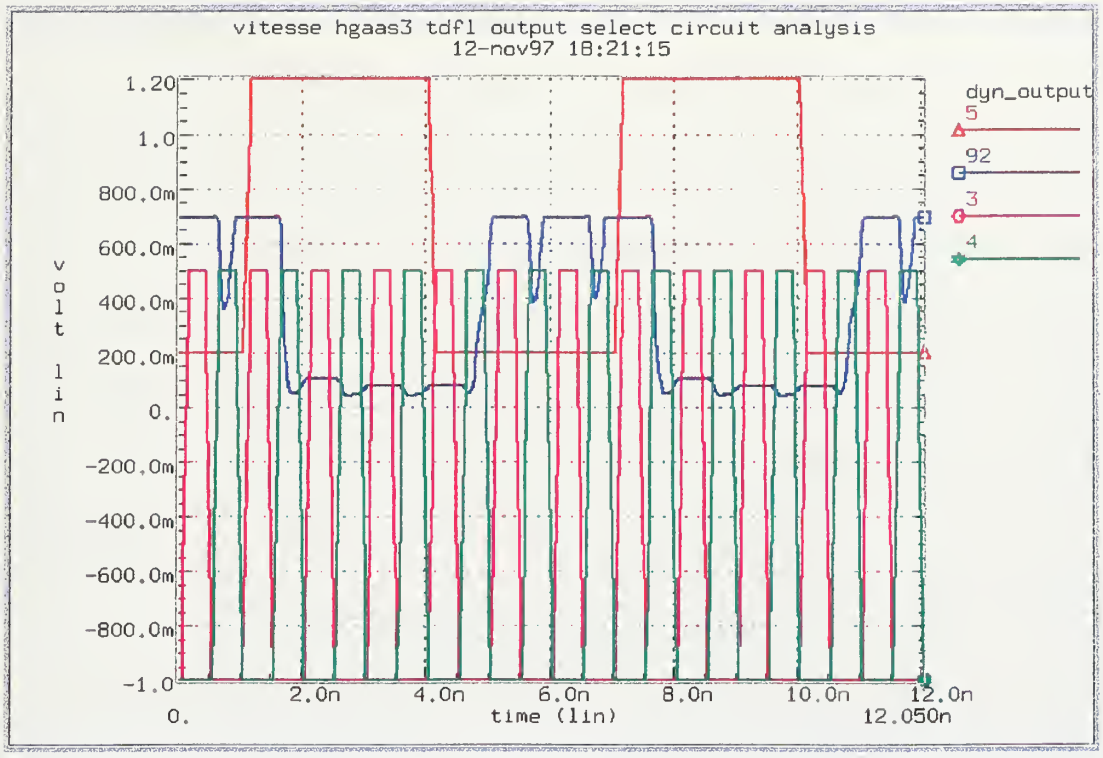


Figure 20. Simulation of TDFL Output Select Circuit at 1 GHz.

3. D Latch Input Select Circuit

A schematic of the TDFL D Latch circuit is given in Figure 21. The D Latch was the most difficult part of the entire design due to the feedback required in the cross-

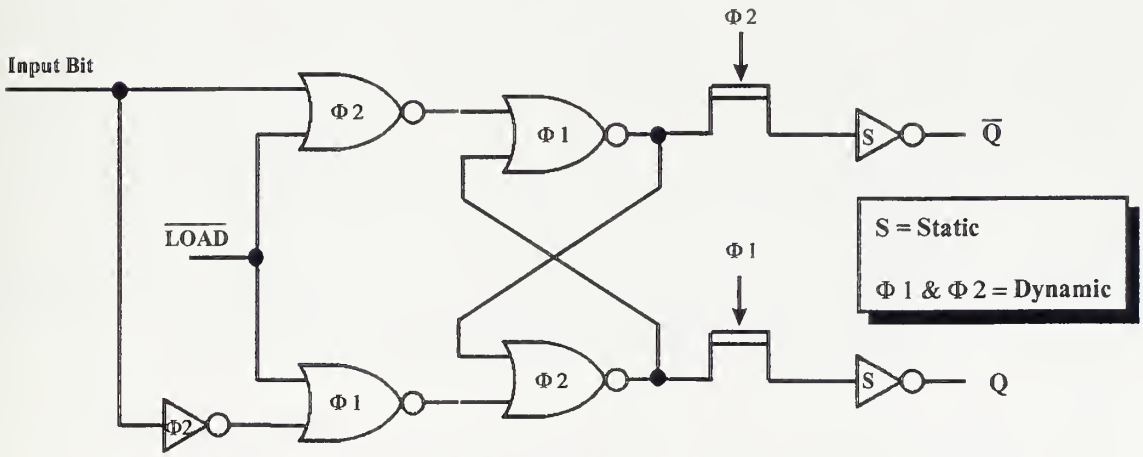


Figure 21. Schematic of TDFL D Latch Circuit.

coupled NOR gates. Several failed attempts were made to redesign the circuit using the pipelined nature of the TDFL topology but the charge sharing and feedback issues proved to be too great. A TDFL gate cannot feed back to itself or to any other gates in its own clock phase. In the case of the D Latch, the complement of the input bit throws the lower set of gates off by one-half clock cycle, which allows the necessary feedback to take place. The drawback to this approach is that the LOAD signal must be active for both clock phases. This was the reason for the static latches on the output select circuit. A simulation of the 3-bit TDFL D Latch at 1 GHz is shown in Figure 22.

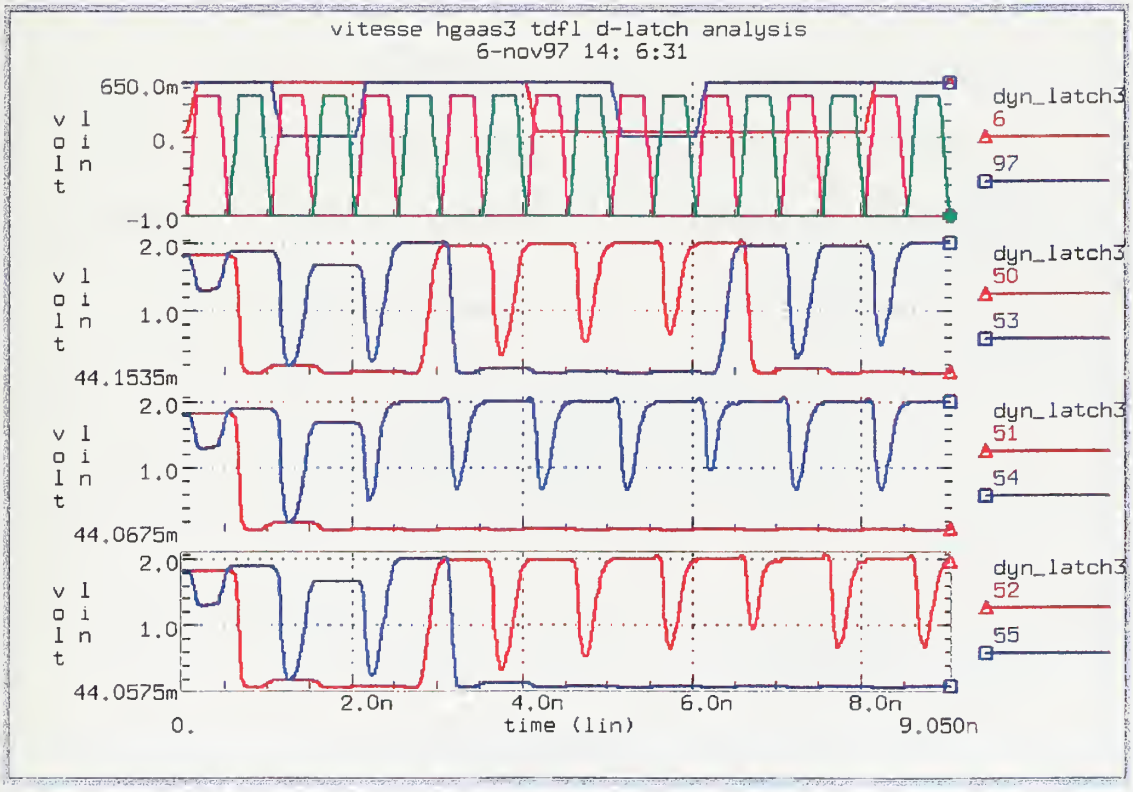


Figure 22. Simulation of TDFL 3-Bit D Latch at 1 GHz.

The four plot windows display the inputs and each of the three output bits with their complements. The top window shows the two clocks, a LOAD signal that is

asserted low at 1 ns and again at 6 ns, and the first input bit that changes from high to low at 4 ns. Input-bits two and three are tied to a low and a high respectively. The outputs show that a 1-0-1 is latched 2 ns after the onset of the LOAD and a 0-0-1 is latched 1.5 ns after the onset of the second load. The transition from low-to-high takes one-half clock cycle longer than from high-to-low. Users of the chip must always wait the longer time period to ensure a proper output.

4. 8-to-1 Multiplexer

The 8-to-1 multiplexer was implemented exactly as the DCFL version, with the addition of a TDFL inverter at the output to complement the output one additional time before latching the output with a SBFL superbuffer. The superbuffer was needed to latch the output so the signal would be valid on both clock phases and match the input bit stream exactly. A schematic of the TDFL 8-to-1 multiplexer is shown in Figure 23.

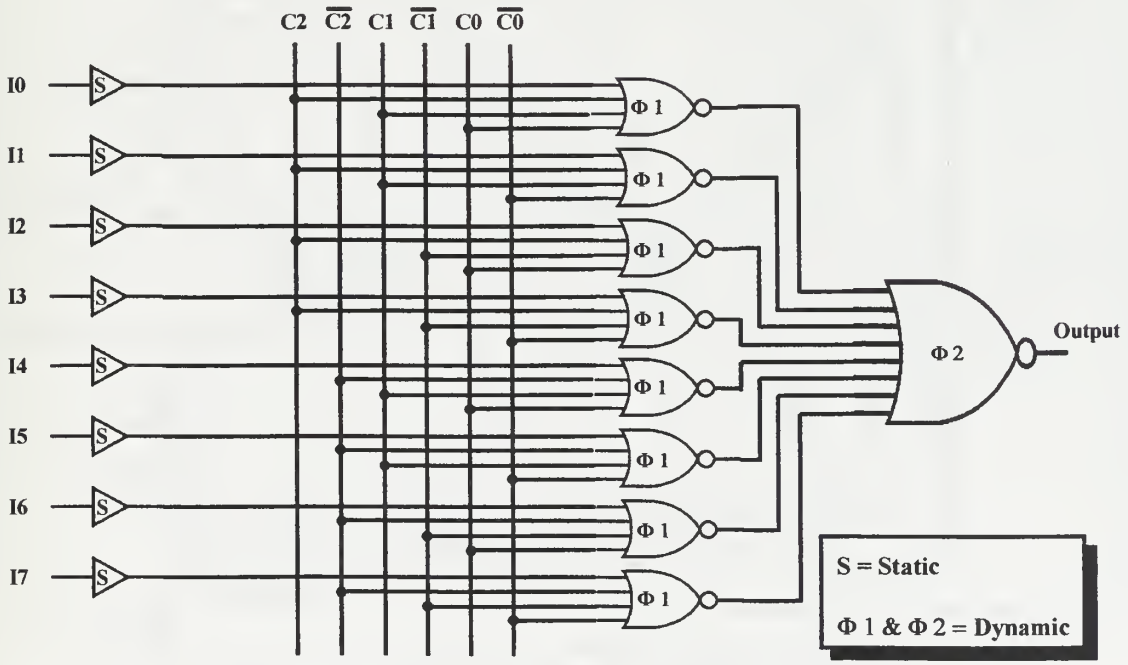


Figure 23. Schematic of TDFL 8-to-1 Multiplexer Circuit.

The simulation for the multiplexer was run first with control signals generated by idealized HSpice sources in order to optimize the gates for a clean output. Simulations were then run again with the designed components connected. After more optimization, a final working design was obtained. Final simulation results follow in Section C.

5. Two-Phase Clock Driver

Because TDFL designs are very sensitive to changes and since two-phase clock drivers are somewhat rare, a self contained clock driver was designed into the TDFL chip. The two-phase clock circuit is derived from a circuit described in Reference [1] and a schematic is shown in Figure 24. The circuit is composed of DCFL and SBFL gates and requires a single-phase clock to operate.

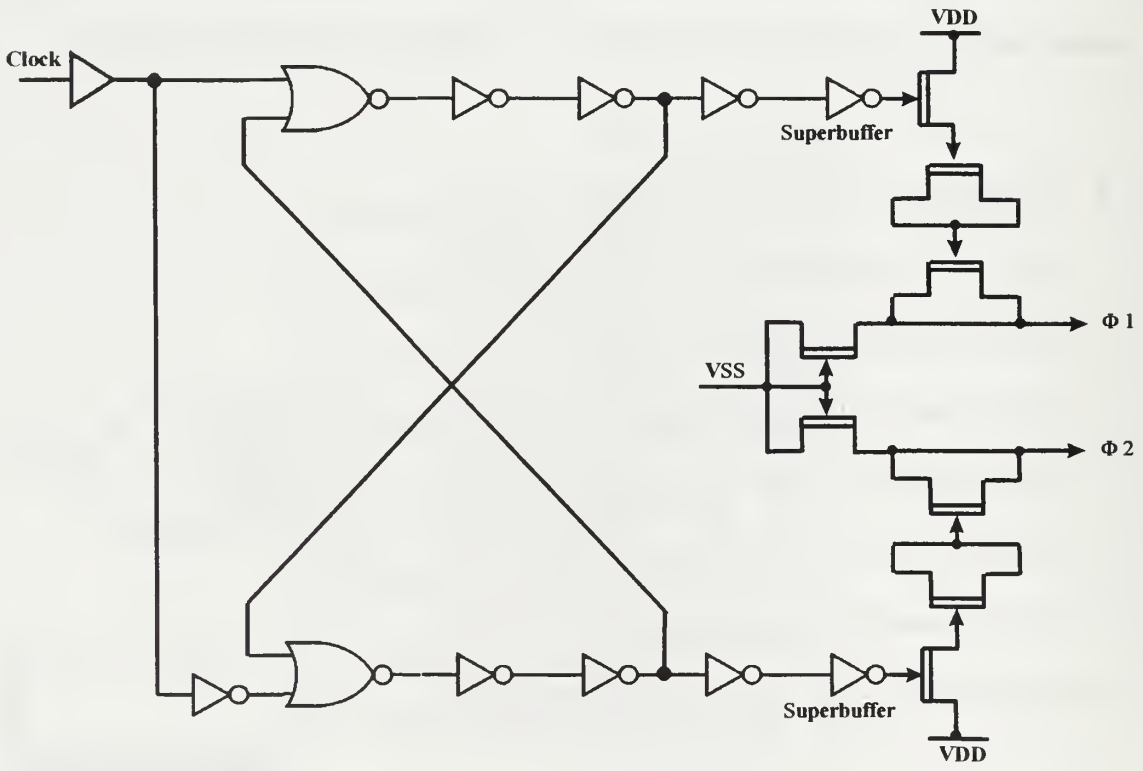


Figure 24. DCFL Two-Phase Clock Generator Circuit.

The cross-coupled NOR gates produce the two-phase clock, with the inverter stages prior to the feedback inducing enough delay to ensure no overlap. The remainder of the transistors are a source-follower circuit that pulls the clock levels down to the desired level of -1.0 V to +0.5 V. In this circuit, V_{DD}=2.0 V and V_{SS}=-2.0 V. Simulation results of the two-phase clock driver are shown in Figure 25.

The top panel shows a 1 GHz single-phase clock pulse both before and after the input receiver. The lower panel shows the two-phase non-overlapping clock signals. The simulation shown is without any loading but the clock driver was tested under varying loads with no appreciable difference in output waveform.

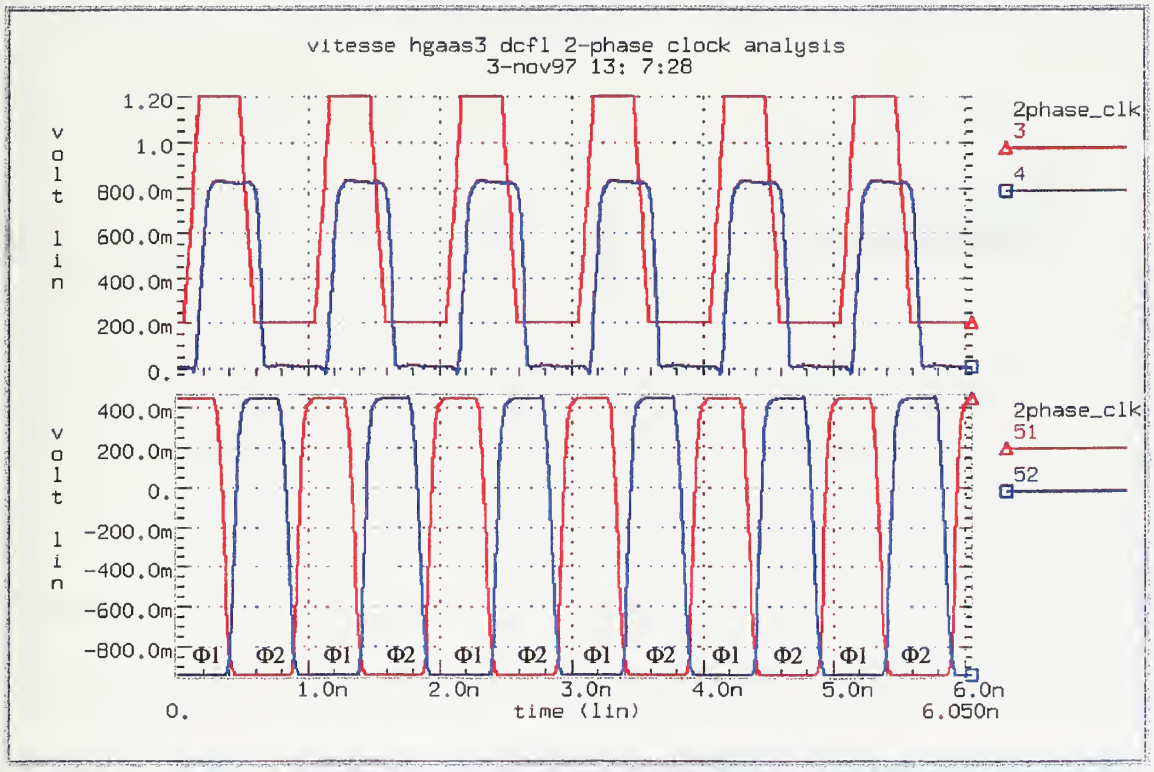


Figure 25. Simulation of DCFL Two-Phase Clock Generator at 1 GHz.

C. SIMULATION OF ENTIRE CROSSPOINT SWITCH

As with the DCFL design, the TDFL crosspoint switch is made up of eight 8-to-1 multiplexers, each controlled by a unique load line from the output select circuit. A schematic of the complete multiplexer, including the additional inverter and pass-gate on the output, is shown in Figure 26.

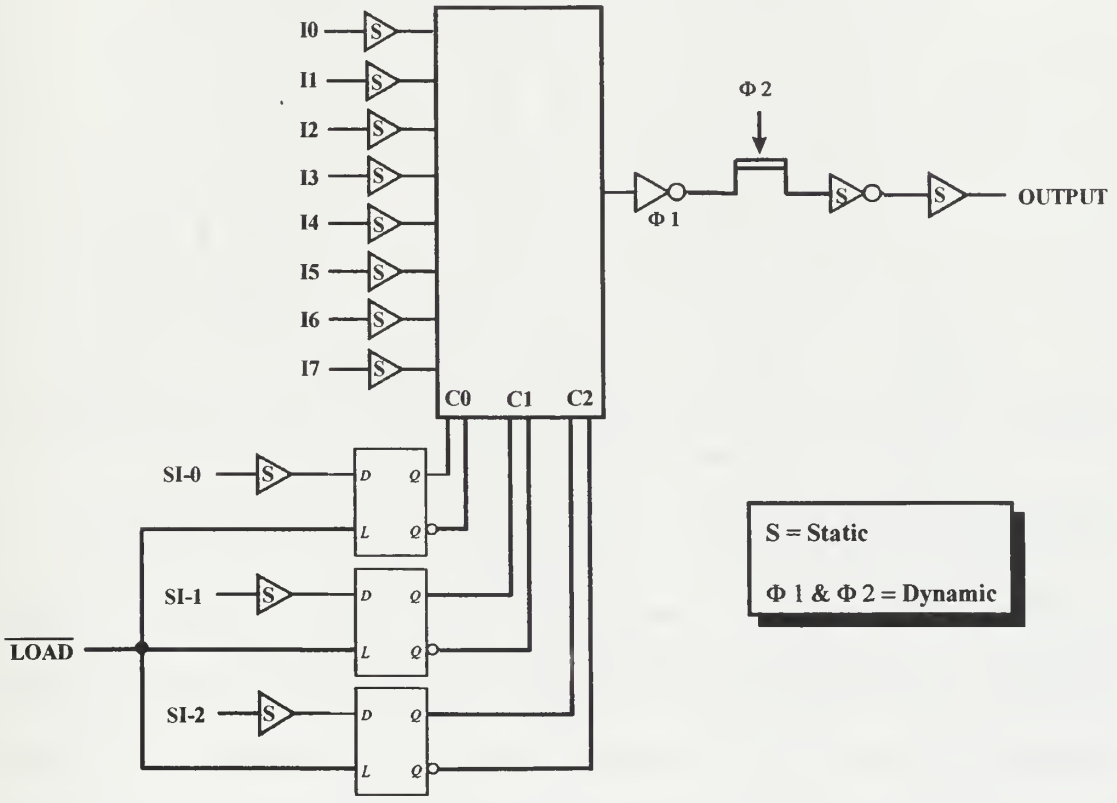


Figure 26. Schematic of Complete TDFL 8-to-1 Multiplexer.

A simulation was performed that included all eight multiplexers and all of the supporting circuitry, including input receivers and output drivers. This complete circuit was tested extensively for proper operation and to examine any difficulties. A simulation of the TDFL crosspoint switch running at the design goal of 1 GHz is shown in Figure 27.

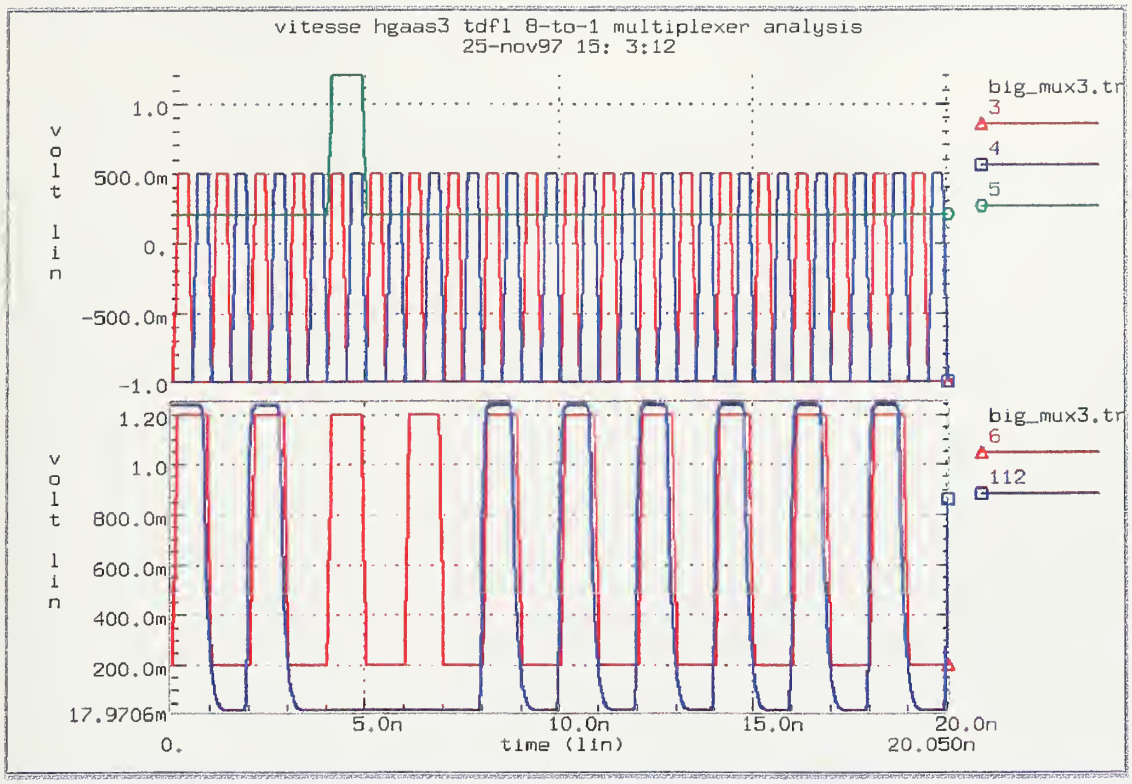


Figure 27. TDFL Crosspoint Switch Simulation at 1 GHz.

The top panel shows the two-phase clock at 1 GHz and the Program Output (PO) signal that is asserted for both clock phases between 4 and 5 ns. Input line 000 and Output line 000 were selected when PO went high. The lower panel shows a 1 Gbps non-return-to-zero (NRZ) bitstream as the input and after a 5-clock cycle delay shows that the output matches the input at 1 Gbps. The 5-clock cycle delay comes from two clock cycles to latch an input line and three clock cycles to propagate through the three TDFL stages in the multiplexer. On high-to-low transitions the output will be valid after only four clock cycles. However, the user must honor the 5-clock cycle delay to get 100% accuracy.

Power consumption figures were also obtained through HSpice simulations. Since the four-input NOR gate was very representative of the other gates in the design, a

simulation was run to investigate the power consumed by a single unloaded gate. The results of the power consumption simulation are shown in Figure 28.

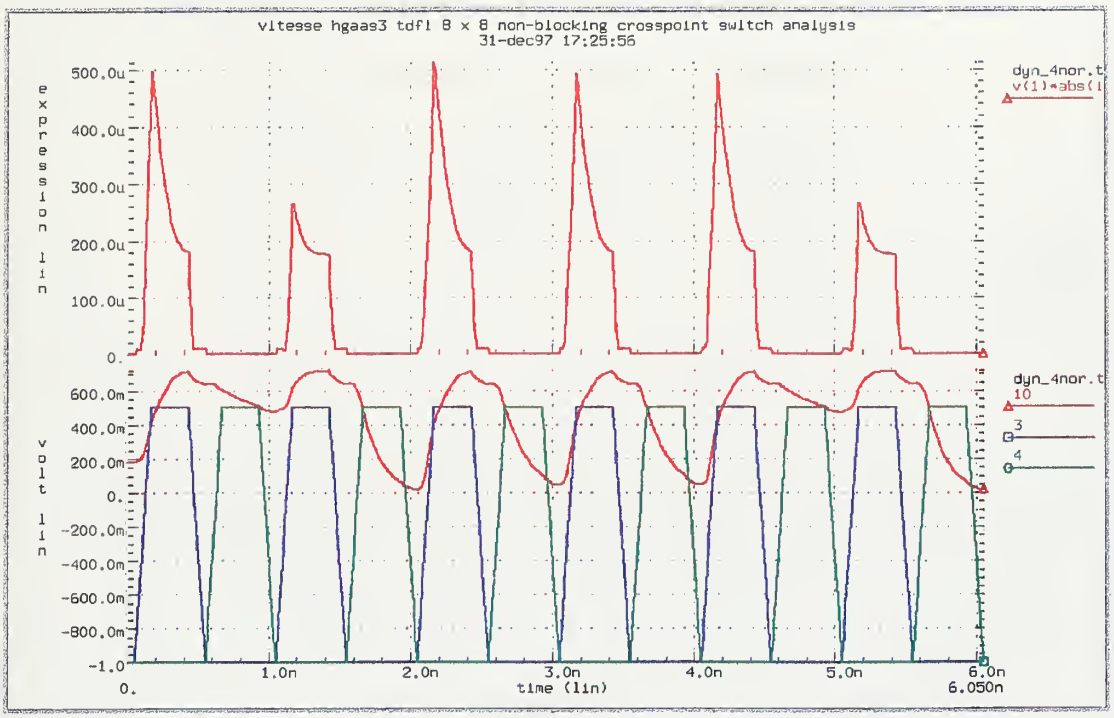


Figure 28. Power Consumption of TDFL 4-Input NOR Gate.

The top panel shows the power consumed by the NOR gate and the bottom panel shows the clock cycles and output level to help explain the results. As described previously, dynamic gates only consume power during clock-level transitions and the simulation confirms this. The NOR gate is only consuming power when in the precharge cycle. Slightly more power is consumed to precharge from a low than from a previous high output. The peak power of this gate is still somewhat high but the average power is quite low. Average power in this example is approximately 100 μW with V_{DD}=2.0 V, as compared to approximately 330 μW for a DCFL NOR gate with equal supply voltage. Power reduction by using the TDFL gate over the DCFL gate is 70%. It is important to

note that with TDFL logic, average power consumption will increase as clock frequency increases.

A simulation was also done to determine the power consumed by the entire crosspoint switch. Figure 29 shows power consumption for the entire chip.

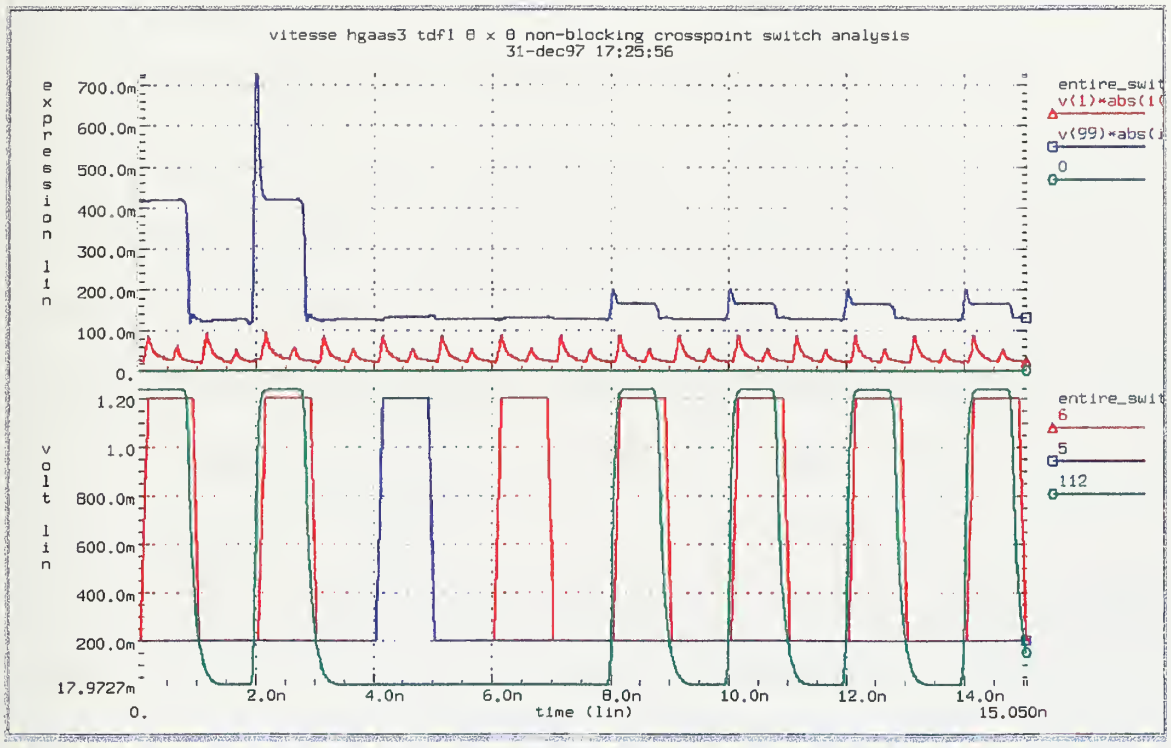


Figure 29. Power Consumption of Entire TDFL Crosspoint Switch.

Separate power supplies were used for the chip core and the input receivers/output drivers. Since the same receivers and drivers were used in the DCFL chip, power consumed in these circuits should be equal. The top panel of the figure shows power, while the lower panel shows the state of operation. The upper plot in the top panel is the power of the input/output pads and is approximately 150 mW, neglecting the spurious peaks at the simulation startup. The lower plot shows the power consumed by the chip core and is approximately 50 mW with a 2.0 V power supply. The 50 mW figure for the

TDFL design is 30% less than the 73 mW figure calculated for the DCFL design. The power savings were less substantial for the complete chip because of the small number of static gates that had to be put into the TDFL design to ensure functionality. A listing of the HSpice code used in the simulations is provided as Appendix A.

D. LAYOUT

The layout of the TDFL crosspoint switch design was the final step in the overall process. Layout of the design was postponed until all other preliminary design steps and simulations were completed. As mentioned previously, the layout was performed with the Magic VLSI design system on a UNIX based host. Magic is a powerful computer-aided design (CAD) package that includes a geometry editor, real-time design rule checker, hierarchy management, circuit extractor, and a foundry interface, all in one intuitive and easy to use program.

The real-time design rule checking feature makes the layout process very rapid with few if any mistakes. Design rule checks were for the Vitesse H-GaAs III process and were contained in the Magic hgaas3.tech technology file. When the entire layout was complete and free of design rule errors, a Caltech Intermediate Format (.CIF) file was created for mask generation by the foundry.

1. DCFL Gates

As described in the detailed design and simulation sections, three DCFL static logic gates were used in the TDFL crosspoint switch design. The DCFL inverter was taken from the H-GaAs III library contained within Magic and is shown in Figure 30.

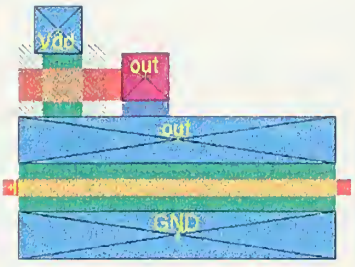


Figure 30. Magic Layout of DCFL Inverter.

Two different types of superbuffers were used in the design. The first is a standard superbuffer that uses EFETs for both pull-up and pull-down but the second type is a modified version of the first that replaces the pull-up EFET with a DFET that allows the output to fully reach V_{DD}. The two types of superbuffers are shown in Figure 31.

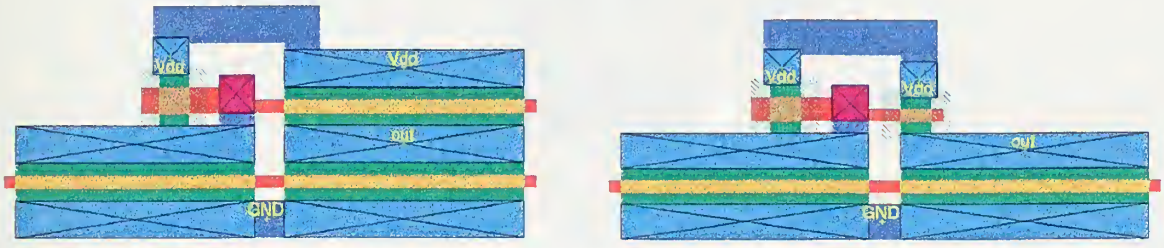


Figure 31. Magic Layout of SBFL Superbuffers.

The clock generator circuit required the use of a library-standard two-input NOR gate, which is shown in Figure 32.

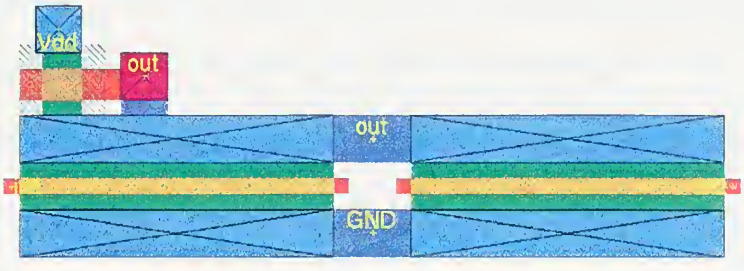


Figure 32. Magic Layout of DCFL 2-Input NOR.

2. TDFL Gates

Since there are no standard TDFL gate libraries and because TDFL gate dimensions vary with circuit location, all TDFL logic gates were drawn from scratch. The TDFL inverter design is shown in Figure 33.

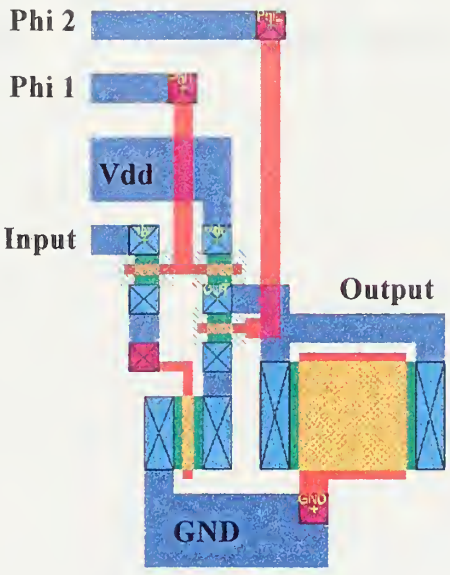


Figure 33. Magic Layout of TDFL Inverter.

The power, ground, and clock lines for all of the TDFL gates were drawn in a bus-like fashion so that arrays of gates could be placed together and many of the connections made automatically. The large square structure in the lower right of the layout is the EFET configured as a reverse-biased diode for increased output capacitance. This particular structure will be very apparent in all of the TDFL gates.

The layout of the TDFL two-input NOR gate follows from the inverter design with the addition of another input pass-transistor and another switching EFET. The four and eight-input NOR gates are simple extensions of the two-input NOR. The TDFL NOR gate family used in the design is shown in Figure 34.

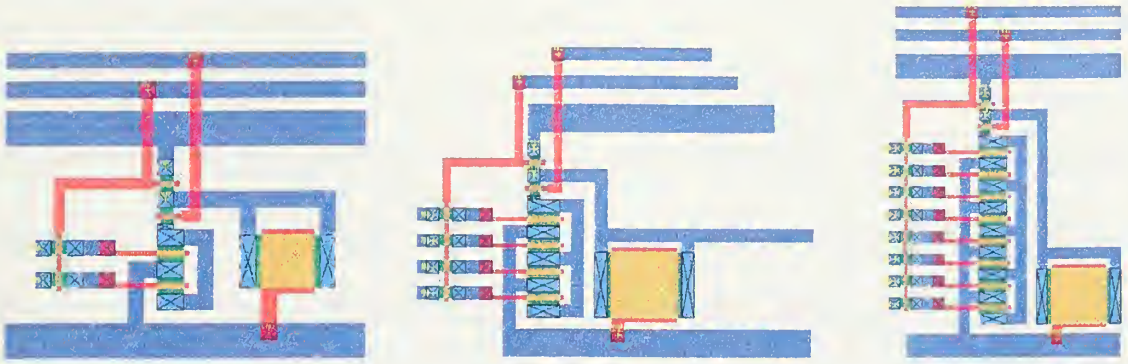


Figure 34. Magic Layouts of TDFL 2, 4 and 8-Input NOR Gates.

Notice the specific clock phase for each of the gates can be easily changed by moving the connections on the two clock bus lines.

3. Higher Level Structures

The output select circuit is shown in Figure 35.

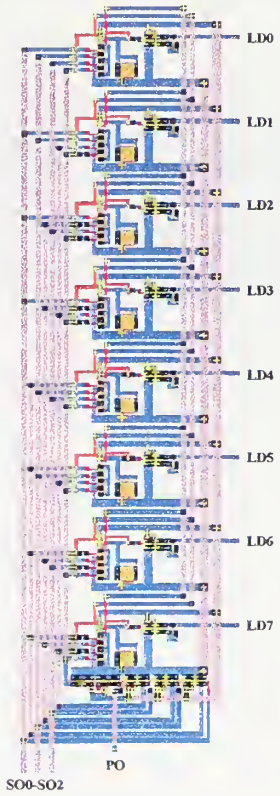


Figure 35. Magic Layout of TDFL Output Select Circuit.

The output select circuit was configured with the eight four-input NOR gates in a single column to simplify distribution of the LOAD signals to the eight multiplexers. The Select Output and Program Output lines enter at the bottom of the structure. The LOAD signals are latched by DCFL inverters and exit on the right-hand side. In the actual chip layout, this structure was flipped horizontally to place the LOAD lines on the left-hand side for easy access to the multiplexers.

The D Latch circuit is shown in Figure 36. The inverter and NOR gates are placed horizontally, automatically creating the power, ground, and clock buses. The input signal enters on the left and the latched output and its complement exit on the right. Three of the latches are connected together to form the 3-bit input select circuit. This circuit was also flipped horizontally when actually placed into the multiplexer circuit.

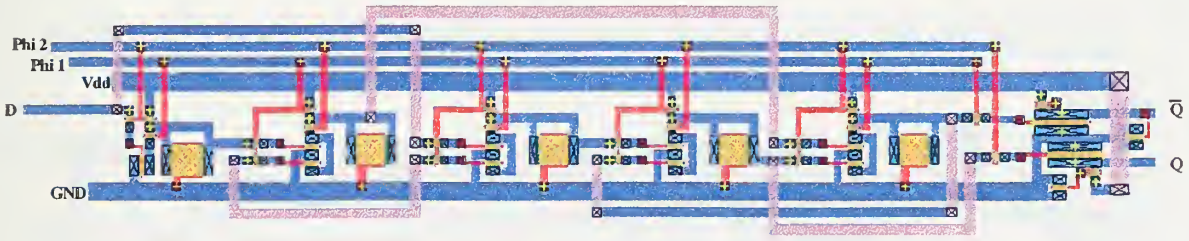


Figure 36. Magic Layout of TDFL D Latch.

A full multiplexer circuit is shown in Figure 37. The 3-bit latch is on the right, the four-input NOR gates are arrayed horizontally in the lower left, and the eight-input NOR gate and its drive circuitry are in the upper left of the figure. When the eight multiplexers are placed in a vertical array, the lines on the right form a vertical bus of all input lines to the multiplexers.

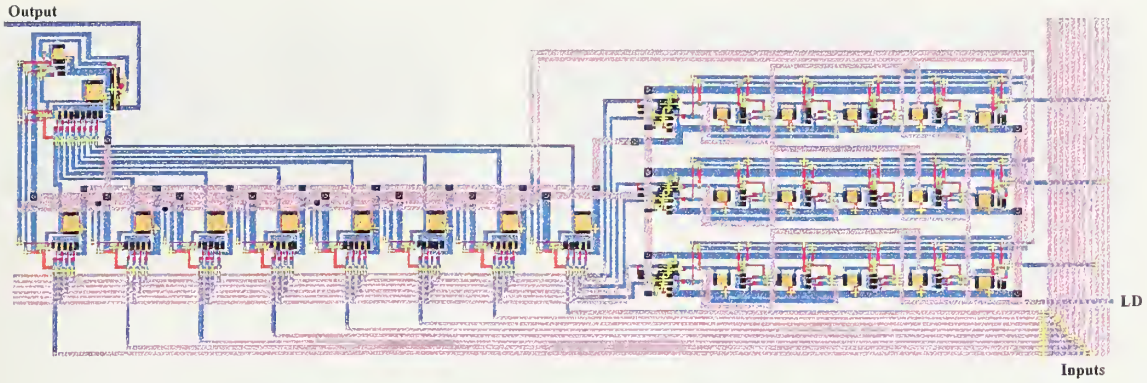


Figure 37. Magic Layout of TDFL 8-to-1 Multiplexer.

The clock generator circuit is shown in Figure 38. The NOR gates and the inverters that form the non-overlapping clock generator are on the left and the level-shifting circuit is on the right. In the full circuit, the clock generator was also flipped horizontally to aid in the routing process. The two-phase clock signals were routed into the center of the chip core and distributed outward in order to minimize clock skew. The clock signal paths are nearly symmetric and the longest run is a maximum of four devices.

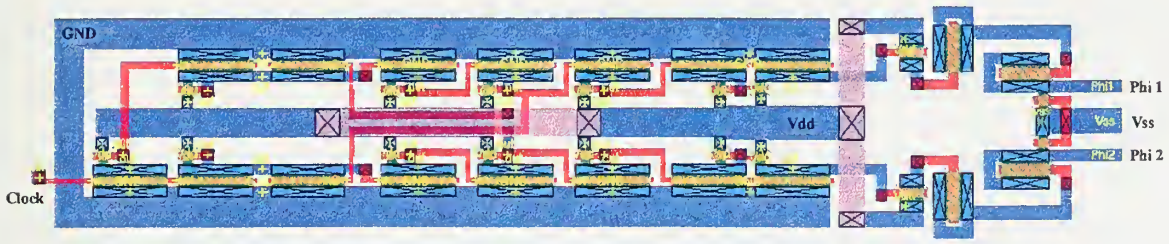


Figure 38. Magic Layout of Two-Phase Clock Generator.

When the entire core was fully connected, a pad ring was built to act as an interface between the chip core and the bonding pads used to connect with the chip package. The input receivers and output drivers were taken from the standard library and are shown in Figure 39.

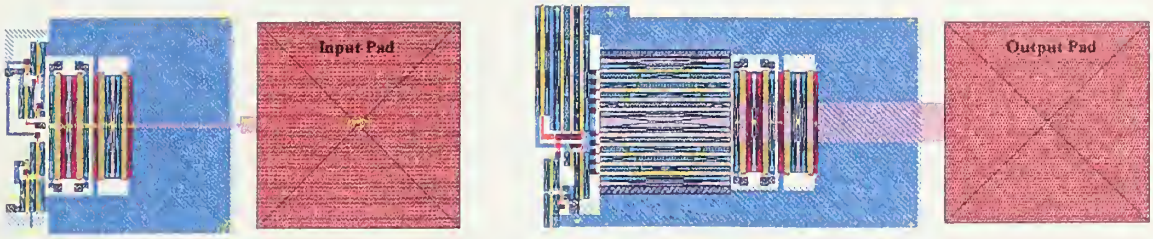


Figure 39. Magic Layouts of Input Receiver and Output Driver.

The input and output pads share power and ground via the pad ring and are fully protected against ESD. The output drivers draw a large amount of peak current. Therefore, numerous power and ground connections are placed around the pad ring to ensure stable voltage levels. Separate power and ground feeds are used for the chip core to further isolate the circuits from the power demands of the pad ring. The clock generator is also attached to the pad ring to keep the amount of power consumed by the core to a minimum.

The entire layout, measured from the edges of the bonding pads, is 3.26 mm by 2.63 mm. Very high device density in the layout was not a design issue as the high number of bonding pads drove the size of the pad ring and ultimately the overall chip size. A full page graphic of the chip and the bonding pad labels is provided as Appendix B. The bonding diagram that shows how the chip is to be connected to the 52-pin package is provided as Appendix C.

V. CONCLUSIONS AND RECOMMENDATIONS

A. SUMMARY OF DESIGN EFFORT

This design effort was conducted to investigate the advantages and disadvantages of using different GaAs MESFET logic topologies to realize a high-speed, low-power, and radiation tolerant crosspoint switch for possible use in future space-based multiprocessor computer systems. GaAs was used in the design because of its advantages over Si in the areas of switching speed and resistance to the effects of ionizing radiation.

Although this research did not involve the design of the DCFL crosspoint switch, enough DCFL logic gates were placed in the TDFL design to get a good feel for working with both static and dynamic logic. In all cases, working with the TDFL topology proved to be more challenging than working with the static DCFL topology. In most cases, designing with DCFL was as simple as going to the standard library of pre-designed gates and picking out the correct one for the task. The DCFL topology is very tolerant of design changes, however, the TDFL topology certainly is not.

When changes are made in a TDFL design, not only do the gate connections need to be changed, but often the basic structure of the primitive logic gates need to be modified. The TDFL logic gates are very sensitive to changes in fan-in and fan-out. The large amount of charge sharing and redistribution that takes place in a TDFL design makes it nearly impossible to construct a set of "standard" logic gates for use throughout a particular design.

The layout of a TDFL circuit is also much more difficult than a similar DCFL circuit due to the addition of the two-phase clocking signals. Along with the clock signal-line routing problem, there is the added complexity of needing two of every TDFL gate so that precharge and evaluate cycles occur on opposite clock phases for successive gates. Another problem that was particularly troublesome in this design was the inability to get a signal and its complement on the same clock phase, due to the induced one-half clock cycle delay. The problem of having an output valid on both clock phases is related and just as troublesome.

Even though designing in the TDFL topology is much more difficult, certain applications could benefit greatly from its advantages. When extremely low power levels are required without sacrificing speed, TDFL is a much better choice than DCFL. Depending on the exact circuits involved, power savings of between 30% and 90% can be expected with TDFL. The other great advantage is in the inherent pipelining found in the TDFL topology. Certain types of sequential logic circuits can be built in a non-standard fashion and achieve full functionality with a reduced number of logic gates.

B. PERFORMANCE ANALYSIS

Since the DCFL crosspoint switch has not been physically tested, and the TDFL crosspoint switch has not even been fabricated, performance analysis must be based upon computer simulation with HSpice.

The TDFL crosspoint switch easily met the design goal of proper operation at a clock speed of 1 GHz and with serial data rates of 1 Gbps. Simulations indicate that the

TDFL crosspoint switch core can run successfully at higher speeds. However, speed for the entire circuit is limited to approximately 1.2 GHz by the input and output drivers used.

While power consumption in the TDFL design was reduced, the reduction was not as great as had been expected. At the individual gate level, the TDFL design beats the DCFL design by approximately 70% but when the entire chip cores are examined, that advantage drops to only about 30%. As explained in Chapter IV, the necessity of using some static gates in the design increased the overall power consumption. Power consumption levels would be even closer together if the clock generator circuit were to be powered from the core of the chip, rather than from the pad ring.

C. RECOMMENDATIONS AND FUTURE WORK

The TDFL topology has both advantages and disadvantages that should be considered by future designers prior to starting a project. The problem should be well defined before selecting either a static or dynamic logic solution. If the design requires ultra-low power consumption and makes use of many sequential circuits, then the added design effort of TDFL might be worth it. On the other hand, if power consumption is not a prime concern, and the circuit to be implemented calls for many signals to be used at the same time as their complements, or for outputs to be valid for both clock periods, then TDFL might not be the best choice.

Several opportunities exist to further the research presented in this thesis. Of course, physical testing of both the fabricated DCFL crosspoint switches and the TDFL crosspoint switches is the next logical step. Testing these circuits can lead to a better

understanding of the capabilities of the logic topologies, as well as serve to check the accuracy of the computer simulation tools.

Since the goal of the project was to design circuits for space-based computer systems, an analysis of the circuits in a radiation environment would also be of great benefit. It would be very interesting to find out if one of the two topologies was more susceptible to radiation effects than the other. A significant difference in radiation tolerance would have a great deal of influence on which logic topology to use in future space-based integrated circuit designs.

LIST OF REFERENCES

1. K. Nary and S. Long, "GaAs Two-Phase Dynamic FET Logic: A Low Power Logic Family for VLSI," IEEE Journal of Solid State Circuits, Vol. 27, No. 10, pp. 1364-1371, Oct. 1992.

2. M. Shur, GaAs Devices and Circuits. New York: Plenum Press, 1987.

3. S. Long and S. Butner, Gallium Arsenide Digital Integrated Circuit Design. New York: McGraw-Hill, 1990.

4. N. Weste and K. Eshraghian, Principles of CMOS VLSI Design, A Systems Perspective, second edition. Addison-Wesley, 1993.

5. R. Pengally, Microwave Field Effect Transistors - Theory, Design and Applications. London: John Wiley & Sons Ltd., 1982.

6. T. Gallander, "Design Tradeoffs in Radiation Hardened ASICs for Space Applications," Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1996.

7. Collective Authors, Meta-Software, HSpice Simulation, Volumes 1, 2 and 3. Meta-Software, Inc., 1992.

8. D. Hoe, "High Speed Gallium Arsenide Dynamic Logic," Doctoral Thesis, University of Toronto, 1991.

9. W. Stallings, Data and Computer Communications, fifth edition. New Jersey: Prentice Hall, 1997.

APPENDIX A. HSPICE FILE USED IN TDFL SWITCH DESIGN

```
Vitesse HGaAs3 TDFL 8 x 8 Non-Blocking Crosspoint Switch Analysis

* include Vitesse HGaAs3 models and parameters for hspice.
.protect
.include '/tools_whitney/cad/meta/parts/vitesse/hgaas3.models'
.lib '/tools_whitney/cad/meta/parts/vitesse/hgaas3.corners' typical
.unprotect

* power supply
vdd 1 0 2.0
vdd2 99 0 2.0  *power supply for receivers and output drivers only
vss 2 0 -2.0

* clock signals
vphi1 3 0 pulse(-1.0V 0.50V 50PS 120PS 120PS 260PS 1000PS)
vphi2 4 0 pulse(-1.0V 0.50V 550PS 120PS 120PS 260PS 1000PS)

*****************************************************************************
* SUB CIRCUITS
*****************************************************************************

* noninverting input receiver
* input-1, output-2, vdd-3, gnd-4
.subckt receiver 2 6 1 0
* ESD protection
j20 1 2 1 0 dp1.2 l=2.0 w=100.0
j21 2 0 2 0 dp1.2 l=2.0 w=100.0
* source follower input stage
j1 1 2 3 3 enh.1 l=0.8 w=16.0
j2 3 0 0 0 dp1.2 l=1.6 w=2.0
* inverting amplifier
j3 1 4 4 0 dp1.2 l=1.6 w=2.0
j4 4 3 0 0 enh.1 l=0.8 w=16.0
* super buffer output stage
j5 1 5 5 0 dp1.2 l=1.6 w=2.0
j6 5 4 0 0 enh.1 l=0.8 w=16.0
j7 1 5 6 0 enh.1 l=0.8 w=16.0
j8 6 4 0 0 enh.1 l=0.8 w=16.0
.ends receiver
```

```
* mininv
* input-1, output-2, vdd-3, vss-4, gnd-5, phi1-6, phi2-7
.subckt mininv 1 2 3 4 5 6 7
j1 3 6 2 4 dp1.1 l=0.8 w=2.0
j2 2 7 8 4 dp1.1 l=0.8 w=2.0
j3 1 6 9 4 dp1.1 l=0.8 w=2.0
j4 8 9 5 5 enh.1 l=0.8 w=6.0
j5 2 5 2 4 enh.2 l=9.0 w=9.0
.ends mininv


* small 2-input NOR
* input-1, input-2, output-3, vdd-4, vss-5, gnd-6, phi1-7, phi2-8
.subckt smallnor2 1 2 3 4 5 6 7 8
j1 4 7 3 5 dp1.1 l=0.8 w=2.0
j2 3 8 9 5 dp1.1 l=0.8 w=2.0
j3 1 7 10 5 dp1.1 l=0.8 w=2.0
j4 9 10 6 6 enh.1 l=0.8 w=4.0
j5 2 7 11 5 dp1.1 l=0.8 w=2.0
j6 9 11 6 6 enh.1 l=0.8 w=4.0
j7 3 6 3 6 enh.2 l=8.0 w=8.0
.ends smallnor2


* small 4-input NOR
* input-1, input-2, input-3, input-4, output-5, vdd-6, vss-7, gnd-8, phi1-9, phi2-10
.subckt smallnor4 1 2 3 4 5 6 7 8 9 10
j1 6 9 5 7 dp1.1 l=0.8 w=2.0
j2 5 10 11 7 dp1.1 l=0.8 w=2.0
j3 1 9 12 7 dp1.1 l=0.8 w=2.0
j4 11 12 8 8 enh.1 l=0.8 w=4.0
j5 2 9 13 7 dp1.1 l=0.8 w=2.0
j6 11 13 8 8 enh.1 l=0.8 w=4.0
j7 3 9 14 7 dp1.1 l=0.8 w=2.0
j8 11 14 8 8 enh.1 l=0.8 w=4.0
j9 4 9 15 7 dp1.1 l=0.8 w=2.0
j10 11 15 8 8 enh.1 l=0.8 w=4.0
j11 5 8 5 8 enh.2 l=9.0 w=9.0
.ends smallnor4
```

```
* medium 4-input NOR
* input-1, input-2, input-3, input-4, output-5, vdd-6, vss-7, gnd-8, phi1-9, phi2-10
.subckt mednor4 1 2 3 4 5 6 7 8 9 10
j1 6 9 5 7 dp1.1 l=0.8 w=2.0
j2 5 10 11 7 dp1.1 l=0.8 w=2.0
j3 1 9 12 7 dp1.1 l=0.8 w=2.0
j4 11 12 8 8 enh.1 l=0.8 w=6.0
j5 2 9 13 7 dp1.1 l=0.8 w=2.0
j6 11 13 8 8 enh.1 l=0.8 w=6.0
j7 3 9 14 7 dp1.1 l=0.8 w=2.0
j8 11 14 8 8 enh.1 l=0.8 w=6.0
j9 4 9 15 7 dp1.1 l=0.8 w=2.0
j10 11 15 8 8 enh.1 l=0.8 w=6.0
j11 5 8 5 8 enh.2 l=12.0 w=12.0
.ends mednor4


* small 8-input NOR
* input-1, input-2, input-3, input-4, input-5, input-6, input-7, input-8,
* output-9, vdd-10, vss-11, gnd-12, phi1-13, phi2-14
.subckt smallnor8 1 2 3 4 5 6 7 8 9 10 11 12 13 14
j1 10 13 9 11 dp1.1 l=0.8 w=2.0
j2 9 14 15 11 dp1.1 l=0.8 w=2.0
j3 1 13 16 11 dp1.1 l=0.8 w=2.0
j4 15 16 12 12 enh.1 l=0.8 w=6.0
j5 2 13 17 11 dp1.1 l=0.8 w=2.0
j6 15 17 12 12 enh.1 l=0.8 w=6.0
j7 3 13 18 11 dp1.1 l=0.8 w=2.0
j8 15 18 12 12 enh.1 l=0.8 w=6.0
j9 4 13 19 11 dp1.1 l=0.8 w=2.0
j10 15 19 12 12 enh.1 l=0.8 w=6.0
j11 5 13 20 11 dp1.1 l=0.8 w=2.0
j12 15 20 12 12 enh.1 l=0.8 w=6.0
j13 6 13 21 11 dp1.1 l=0.8 w=2.0
j14 15 21 12 12 enh.1 l=0.8 w=6.0
j15 7 13 22 11 dp1.1 l=0.8 w=2.0
j16 15 22 12 12 enh.1 l=0.8 w=6.0
j17 8 13 23 11 dp1.1 l=0.8 w=2.0
j18 15 23 12 12 enh.1 l=0.8 w=6.0
j19 9 12 9 12 enh.2 l=12.0 w=12.0
.ends smallnor8
```

```
* static inverter
* input-1, output-2, vdd-3, gnd-4
.subckt statinv 1 2 3 4
j1 3 2 2 4 dp1.2 l=1.6 w=2.0
j2 2 1 4 4 enh.1 l=0.8 w=16.0
.ends statinv


* static superbuffer
* input-1, output-2, vdd-3, gnd-4
.subckt superbuffer 1 2 3 4
j1 3 5 5 4 dp1.2 l=1.6 w=2.0
j2 5 1 4 4 enh.1 l=0.8 w=16.0
j3 3 5 2 4 dp1.1 l=0.8 w=2.0
j4 2 1 4 4 enh.1 l=0.8 w=16.0
.ends superbuffer


* Non-inverting Output Driver
* input-1, output-2, vdd-3, gnd-4
.subckt driver 2 7 1 0
* super buffer input stages
j1 1 3 3 0 dp1.2 l=1.6 w=2.0
j2 3 2 0 0 enh.1 l=0.8 w=16.0
j3 1 3 4 0 dp1.1 l=0.8 w=2.0
j4 4 2 0 0 enh.1 l=0.8 w=16.0
j5 1 5 5 0 dp1.1 l=0.8 w=6.8
j6 5 4 0 0 enh.1 l=0.8 w=54.4
j7 1 5 6 0 dp1.1 l=0.8 w=23.2
j8 6 4 0 0 enh.1 l=0.8 w=185.6
* source follower output
j9 1 6 7 0 enh.1 l=0.8 w=631.2
* ESD protection
j10 1 7 1 0 dp1.2 l=2.0 w=100.0
j11 7 0 7 0 dp1.2 l=2.0 w=100.0
* load
r1 7 8 50
c1 7 8 2pf
vtest1 0 8 0.0
.ends driver
```

```
*************************************************************************
* input signals
*************************************************************************
vin1 5 0 pulse(0.2V 1.2V 4050PS 120PS 120PS 760PS 20000PS)
vin2 6 0 pulse(0.2V 1.2V 50PS 120PS 120PS 760PS 2000PS)
vin3 7 0 pulse(1.2V 0.2V 50PS 120PS 120PS 3760PS 10000PS)
vlow 8 0 0.2
vhigh 9 0 1.2

*************************************************************************
* control signals
*************************************************************************
vp0 10 0 1.2
vso0 11 0 0.2
vso1 12 0 0.2
vso2 13 0 0.2
vsi0 14 0 0.2
vsi1 15 0 0.2
vsi2 16 0 0.2

*************************************************************************
* main circuit
*************************************************************************

* buffered input signals
x1 5 17 99 0 receiver          *PO
x2 17 18 1 0 superbuffer       *BPO
*output selects
x3 11 19 99 0 receiver         *DC0
x4 19 20 1 0 statinv           *BDC0
x5 12 21 99 0 receiver                 *DC1
x6 21 22 1 0 statinv                   *BDC1
x7 13 23 99 0 receiver         *DC2
x8 23 24 1 0 statinv           *BDC2
*input selects
x9 14 25 99 0 receiver         *MC0
j29 0 25 0 0 enh.1 l=0.8 w=4.0                 *Load
x10 15 26 99 0 receiver                *MC1
j30 0 26 0 0 enh.1 l=0.8 w=4.0                 *Load
x11 16 27 99 0 receiver                *MC2
j31 0 27 0 0 enh.1 l=0.8 w=4.0                 *Load
```

```
*****************************************************************************
*Select Mux (active low load signal)
*****************************************************************************
x12 23 21 19 18 30 1 2 0 3 4 smallnor4
j1 30 4 40 2 dp1.1 l=1.6 w=2.0
x13 40 50 1 0 statinv                   *BLD0
j2 0 50 0 0 enh.1 l=0.8 w=4.0           *Load
x14 23 21 20 18 31 1 2 0 3 4 smallnor4
j3 31 4 41 2 dp1.1 l=1.6 w=2.0
x15 41 51 1 0 statinv                   *BLD1
j4 0 51 0 0 enh.1 l=0.8 w=4.0           *Load
x16 23 22 19 18 32 1 2 0 3 4 smallnor4
j5 32 4 42 2 dp1.1 l=1.6 w=2.0
x17 42 52 1 0 statinv                   *BLD2
j6 0 52 0 0 enh.1 l=0.8 w=4.0           *Load
x18 23 22 20 18 33 1 2 0 3 4 smallnor4
j7 33 4 43 2 dp1.1 l=1.6 w=2.0
x19 43 53 1 0 statinv                   *BLD3
j8 0 53 0 0 enh.1 l=0.8 w=4.0           *Load
x20 24 21 19 18 34 1 2 0 3 4 smallnor4
j9 34 4 44 2 dp1.1 l=1.6 w=2.0
x21 44 54 1 0 statinv                   *BLD4
j10 0 54 0 0 enh.1 l=0.8 w=4.0                  *Load
x22 24 21 20 18 35 1 2 0 3 4 smallnor4
j11 35 4 45 2 dp1.1 l=1.6 w=2.0
x23 45 55 1 0 statinv                   *BLD5
j12 0 55 0 0 enh.1 l=0.8 w=4.0                  *Load
x24 24 22 19 18 36 1 2 0 3 4 smallnor4
j13 36 4 46 2 dp1.1 l=1.6 w=2.0
x25 46 56 1 0 statinv                   *BLD6
j14 0 56 0 0 enh.1 l=0.8 w=4.0                  *Load
x26 24 22 20 18 37 1 2 0 3 4 smallnor4
j15 37 4 47 2 dp1.1 l=1.6 w=2.0
x27 47 57 1 0 statinv                   *BLD7
j16 0 57 0 0 enh.1 l=0.8 w=4.0                  *Load
```

```
*************************************************************************
*Latches (Input Line Select)
*************************************************************************
*Latches for Mux #0
x30 25 60 1 2 0 4 3 mininv
x31 25 50 61 1 2 0 4 3 smallnor2
x32 60 50 62 1 2 0 3 4 smallnor2
x33 61 64 63 1 2 0 3 4 smallnor2
x34 62 63 64 1 2 0 4 3 smallnor2
j17 63 4 65 2 dp1.1 l=1.6 w=2.0
x35 65 67 1 0 statinv                *BC0
j18 0 67 0 0 enh.1 l=0.8 w=4.0               *Load
j19 64 3 66 2 dp1.1 l=1.6 w=2.0
x36 66 68 1 0 statinv                *C0
j20 0 68 0 0 enh.1 l=0.8 w=4.0               *Load

x40 26 70 1 2 0 4 3 mininv
x41 26 50 71 1 2 0 4 3 smallnor2
x42 70 50 72 1 2 0 3 4 smallnor2
x43 71 74 73 1 2 0 3 4 smallnor2
x44 72 73 74 1 2 0 4 3 smallnor2
j21 73 4 75 2 dp1.1 l=1.6 w=2:0
x45 75 77 1 0 statinv                *BC1
j22 0 77 0 0 enh.1 l=0.8 w=4.0               *Load
j23 74 3 76 2 dp1.1 l=1.6 w=2.0
x46 76 78 1 0 statinv                *C1
j24 0 78 0 0 enh.1 l=0.8 w=4.0               *Load

x50 27 80 1 2 0 4 3 mininv
x51 27 50 81 1 2 0 4 3 smallnor2
x52 80 50 82 1 2 0 3 4 smallnor2
x53 81 84 83 1 2 0 3 4 smallnor2
x54 82 83 84 1 2 0 4 3 smallnor2
j25 83 4 85 2 dp1.1 l=1.6 w=2.0
x55 85 87 1 0 statinv                *BC2
j26 0 87 0 0 enh.1 l=0.8 w=4.0               *Load
j27 84 3 86 2 dp1.1 l=1.6 w=2.0
x56 86 88 1 0 statinv                *C2
j28 0 88 0 0 enh.1 l=0.8 w=4.0               *Load

*Latches for Mux #1
x130 25 160 1 2 0 4 3 mininv
x131 25 51 161 1 2 0 4 3 smallnor2
x132 160 51 162 1 2 0 3 4 smallnor2
```

```
x133 161 164 163 1 2 0 3 4 smallnor2
x134 162 163 164 1 2 0 4 3 smallnor2
j117 163 4 165 2 dp1.1 l=1.6 w=2.0
x135 165 167 1 0 statinv                *BC0
j118 0 167 0 0 enh.1 l=0.8 w=4.0        *Load
j119 164 3 166 2 dp1.1 l=1.6 w=2.0
x136 166 168 1 0 statinv                *C0
j120 0 168 0 0 enh.1 l=0.8 w=4.0        *Load

x140 26 170 1 2 0 4 3 mininv
x141 26 51 171 1 2 0 4 3 smallnor2
x142 170 51 172 1 2 0 3 4 smallnor2
x143 171 174 173 1 2 0 3 4 smallnor2
x144 172 173 174 1 2 0 4 3 smallnor2
j121 173 4 175 2 dp1.1 l=1.6 w=2.0
x145 175 177 1 0 statinv                *BC1
j122 0 177 0 0 enh.1 l=0.8 w=4.0        *Load
j123 174 3 176 2 dp1.1 l=1.6 w=2.0
x146 176 178 1 0 statinv                *C1
j124 0 178 0 0 enh.1 l=0.8 w=4.0        *Load

x150 27 180 1 2 0 4 3 mininv
x151 27 51 181 1 2 0 4 3 smallnor2
x152 180 51 182 1 2 0 3 4 smallnor2
x153 181 184 183 1 2 0 3 4 smallnor2
x154 182 183 184 1 2 0 4 3 smallnor2
j125 183 4 185 2 dp1.1 l=1.6 w=2.0
x155 185 187 1 0 statinv                *BC2
j126 0 187 0 0 enh.1 l=0.8 w=4.0        *Load
j127 184 3 186 2 dp1.1 l=1.6 w=2.0
x156 186 188 1 0 statinv                *C2
j128 0 188 0 0 enh.1 l=0.8 w=4.0        *Load

*Latches for Mux #2
x230 25 260 1 2 0 4 3 mininv
x231 25 52 261 1 2 0 4 3 smallnor2
x232 260 52 262 1 2 0 3 4 smallnor2
x233 261 264 263 1 2 0 3 4 smallnor2
x234 262 263 264 1 2 0 4 3 smallnor2
j217 263 4 265 2 dp1.1 l=1.6 w=2.0
x235 265 267 1 0 statinv                *BC0
j218 0 267 0 0 enh.1 l=0.8 w=4.0        *Load
j219 264 3 266 2 dp1.1 l=1.6 w=2.0
x236 266 268 1 0 statinv                *C0
```

```
j220 0 268 0 0 enh.1 l=0.8 w=4.0          *Load

x240 26 270 1 2 0 4 3 mininv
x241 26 52 271 1 2 0 4 3 smallnor2
x242 270 52 272 1 2 0 3 4 smallnor2
x243 271 274 273 1 2 0 3 4 smallnor2
x244 272 273 274 1 2 0 4 3 smallnor2
j221 273 4 275 2 dp1.1 l=1.6 w=2.0
x245 275 277 1 0 statinv                   *BC1
j222 0 277 0 0 enh.1 l=0.8 w=4.0           *Load
j223 274 3 276 2 dp1.1 l=1.6 w=2.0
x246 276 278 1 0 statinv                   *C1
j224 0 278 0 0 enh.1 l=0.8 w=4.0           *Load

x250 27 280 1 2 0 4 3 mininv
x251 27 52 281 1 2 0 4 3 smallnor2
x252 280 52 282 1 2 0 3 4 smallnor2
x253 281 284 283 1 2 0 3 4 smallnor2
x254 282 283 284 1 2 0 4 3 smallnor2
j225 283 4 285 2 dp1.1 l=1.6 w=2.0
x255 285 287 1 0 statinv                   *BC2
j226 0 287 0 0 enh.1 l=0.8 w=4.0           *Load
j227 284 3 286 2 dp1.1 l=1.6 w=2.0
x256 286 288 1 0 statinv                   *C2
j228 0 288 0 0 enh.1 l=0.8 w=4.0           *Load

*Latches for Mux #3
x330 25 360 1 2 0 4 3 mininv
x331 25 53 361 1 2 0 4 3 smallnor2
x332 360 53 362 1 2 0 3 4 smallnor2
x333 361 364 363 1 2 0 3 4 smallnor2
x334 362 363 364 1 2 0 4 3 smallnor2
j317 363 4 365 2 dp1.1 l=1.6 w=2.0
x335 365 367 1 0 statinv                   *BC0
j318 0 367 0 0 enh.1 l=0.8 w=4.0           *Load
j319 364 3 366 2 dp1.1 l=1.6 w=2.0
x336 366 368 1 0 statinv                   *C0
j320 0 368 0 0 enh.1 l=0.8 w=4.0           *Load

x340 26 370 1 2 0 4 3 mininv
x341 26 53 371 1 2 0 4 3 smallnor2
x342 370 53 372 1 2 0 3 4 smallnor2
x343 371 374 373 1 2 0 3 4 smallnor2
x344 372 373 374 1 2 0 4 3 smallnor2
```

```
j321 373 4 375 2 dp1.1 l=1.6 w=2.0
x345 375 377 1 0 statinv                *BC1
j322 0 377 0 0 enh.1 l=0.8 w=4.0        *Load
j323 374 3 376 2 dp1.1 l=1.6 w=2.0
x346 376 378 1 0 statinv                *C1
j324 0 378 0 0 enh.1 l=0.8 w=4.0        *Load

x350 27 380 1 2 0 4 3 mininv
x351 27 53 381 1 2 0 4 3 smallnor2
x352 380 53 382 1 2 0 3 4 smallnor2
x353 381 384 383 1 2 0 3 4 smallnor2
x354 382 383 384 1 2 0 4 3 smallnor2
j325 383 4 385 2 dp1.1 l=1.6 w=2.0
x355 385 387 1 0 statinv                *BC2
j326 0 387 0 0 enh.1 l=0.8 w=4.0        *Load
j327 384 3 386 2 dp1.1 l=1.6 w=2.0
x356 386 388 1 0 statinv                *C2
j328 0 388 0 0 enh.1 l=0.8 w=4.0        *Load

*Latches for Mux #4
x430 25 460 1 2 0 4 3 mininv
x431 25 54 461 1 2 0 4 3 smallnor2
x432 460 54 462 1 2 0 3 4 smallnor2
x433 461 464 463 1 2 0 3 4 smallnor2
x434 462 463 464 1 2 0 4 3 smallnor2
j417 463 4 465 2 dp1.1 l=1.6 w=2.0
x435 465 467 1 0 statinv                *BC0
j418 0 467 0 0 enh.1 l=0.8 w=4.0        *Load
j419 464 3 466 2 dp1.1 l=1.6 w=2.0
x436 466 468 1 0 statinv                *C0
j420 0 468 0 0 enh.1 l=0.8 w=4.0        *Load

x440 26 470 1 2 0 4 3 mininv
x441 26 54 471 1 2 0 4 3 smallnor2
x442 470 54 472 1 2 0 3 4 smallnor2
x443 471 474 473 1 2 0 3 4 smallnor2
x444 472 473 474 1 2 0 4 3 smallnor2
j421 473 4 475 2 dp1.1 l=1.6 w=2.0
x445 475 477 1 0 statinv                *BC1
j422 0 477 0 0 enh.1 l=0.8 w=4.0        *Load
j423 474 3 476 2 dp1.1 l=1.6 w=2.0
x446 476 478 1 0 statinv                *C1
j424 0 478 0 0 enh.1 l=0.8 w=4.0        *Load
```

```
x450 27 480 1 2 0 4 3 mininv
x451 27 54 481 1 2 0 4 3 smallnor2
x452 480 54 482 1 2 0 3 4 smallnor2
x453 481 484 483 1 2 0 3 4 smallnor2
x454 482 483 484 1 2 0 4 3 smallnor2
j425 483 4 485 2 dp1.1 l=1.6 w=2.0
x455 485 487 1 0 statinv                *BC2
j426 0 487 0 0 enh.1 l=0.8 w=4.0        *Load
j427 484 3 486 2 dp1.1 l=1.6 w=2.0
x456 486 488 1 0 statinv                *C2
j428 0 488 0 0 enh.1 l=0.8 w=4.0        *Load

*Latches for Mux #5
x530 25 560 1 2 0 4 3 mininv
x531 25 55 561 1 2 0 4 3 smallnor2
x532 560 55 562 1 2 0 3 4 smallnor2
x533 561 564 563 1 2 0 3 4 smallnor2
x534 562 563 564 1 2 0 4 3 smallnor2
j517 563 4 565 2 dp1.1 l=1.6 w=2.0
x535 565 567 1 0 statinv                *BC0
j518 0 567 0 0 enh.1 l=0.8 w=4.0        *Load
j519 564 3 566 2 dp1.1 l=1.6 w=2.0
x536 566 568 1 0 statinv                *C0
j520 0 568 0 0 enh.1 l=0.8 w=4.0        *Load

x540 26 570 1 2 0 4 3 mininv
x541 26 55 571 1 2 0 4 3 smallnor2
x542 570 55 572 1 2 0 3 4 smallnor2
x543 571 574 573 1 2 0 3 4 smallnor2
x544 572 573 574 1 2 0 4 3 smallnor2
j521 573 4 575 2 dp1.1 l=1.6 w=2.0
x545 575 577 1 0 statinv                *BC1
j522 0 577 0 0 enh.1 l=0.8 w=4.0        *Load
j523 574 3 576 2 dp1.1 l=1.6 w=2.0
x546 576 578 1 0 statinv                *C1
j524 0 578 0 0 enh.1 l=0.8 w=4.0        *Load

x550 27 580 1 2 0 4 3 mininv
x551 27 55 581 1 2 0 4 3 smallnor2
x552 580 55 582 1 2 0 3 4 smallnor2
x553 581 584 583 1 2 0 3 4 smallnor2
x554 582 583 584 1 2 0 4 3 smallnor2
j525 583 4 585 2 dp1.1 l=1.6 w=2.0
x555 585 587 1 0 statinv                *BC2
```

```
j526 0 587 0 0 enh.1 l=0.8 w=4.0          *Load
j527 584 3 586 2 dp1.1 l=1.6 w=2.0
x556 586 588 1 0 statinv                  *C2
j528 0 588 0 0 enh.1 l=0.8 w=4.0          *Load

*Latches for Mux #6
x630 25 660 1 2 0 4 3 mininv
x631 25 56 661 1 2 0 4 3 smallnor2
x632 660 56 662 1 2 0 3 4 smallnor2
x633 661 664 663 1 2 0 3 4 smallnor2
x634 662 663 664 1 2 0 4 3 smallnor2
j617 663 4 665 2 dp1.1 l=1.6 w=2.0
x635 665 667 1 0 statinv                  *BC0
j618 0 667 0 0 enh.1 l=0.8 w=4.0          *Load
j619 664 3 666 2 dp1.1 l=1.6 w=2.0
x636 666 668 1 0 statinv                  *C0
j620 0 668 0 0 enh.1 l=0.8 w=4.0          *Load

x640 26 670 1 2 0 4 3 mininv
x641 26 56 671 1 2 0 4 3 smallnor2
x642 670 56 672 1 2 0 3 4 smallnor2
x643 671 674 673 1 2 0 3 4 smallnor2
x644 672 673 674 1 2 0 4 3 smallnor2
j621 673 4 675 2 dp1.1 l=1.6 w=2.0
x645 675 677 1 0 statinv                  *BC1
j622 0 677 0 0 enh.1 l=0.8 w=4.0          *Load
j623 674 3 676 2 dp1.1 l=1.6 w=2.0
x646 676 678 1 0 statinv                  *C1
j624 0 678 0 0 enh.1 l=0.8 w=4.0          *Load

x650 27 680 1 2 0 4 3 mininv
x651 27 56 681 1 2 0 4 3 smallnor2
x652 680 56 682 1 2 0 3 4 smallnor2
x653 681 684 683 1 2 0 3 4 smallnor2
x654 682 683 684 1 2 0 4 3 smallnor2
j625 683 4 685 2 dp1.1 l=1.6 w=2.0
x655 685 687 1 0 statinv                  *BC2
j626 0 687 0 0 enh.1 l=0.8 w=4.0          *Load
j627 684 3 686 2 dp1.1 l=1.6 w=2.0
x656 686 688 1 0 statinv                  *C2
j628 0 688 0 0 enh.1 l=0.8 w=4.0          *Load

*Latches for Mux #7
x730 25 760 1 2 0 4 3 mininv
```

```
x731 25 57 761 1 2 0 4 3 smallnor2
x732 760 57 762 1 2 0 3 4 smallnor2
x733 761 764 763 1 2 0 3 4 smallnor2
x734 762 763 764 1 2 0 4 3 smallnor2
j717 763 4 765 2 dp1.1 l=1.6 w=2.0
x735 765 767 1 0 statinv                *BC0
j718 0 767 0 0 enh.1 l=0.8 w=4.0        *Load
j719 764 3 766 2 dp1.1 l=1.6 w=2.0
x736 766 768 1 0 statinv                *C0
j720 0 768 0 0 enh.1 l=0.8 w=4.0        *Load

x740 26 770 1 2 0 4 3 mininv
x741 26 57 771 1 2 0 4 3 smallnor2
x742 770 57 772 1 2 0 3 4 smallnor2
x743 771 774 773 1 2 0 3 4 smallnor2
x744 772 773 774 1 2 0 4 3 smallnor2
j721 773 4 775 2 dp1.1 l=1.6 w=2.0
x745 775 777 1 0 statinv                *BC1
j722 0 777 0 0 enh.1 l=0.8 w=4.0        *Load
j723 774 3 776 2 dp1.1 l=1.6 w=2.0
x746 776 778 1 0 statinv                *C1
j724 0 778 0 0 enh.1 l=0.8 w=4.0        *Load

x750 27 780 1 2 0 4 3 mininv
x751 27 57 781 1 2 0 4 3 smallnor2
x752 780 57 782 1 2 0 3 4 smallnor2
x753 781 784 783 1 2 0 3 4 smallnor2
x754 782 783 784 1 2 0 4 3 smallnor2
j725 783 4 785 2 dp1.1 l=1.6 w=2.0
x755 785 787 1 0 statinv                *BC2
j726 0 787 0 0 enh.1 l=0.8 w=4.0        *Load
j727 784 3 786 2 dp1.1 l=1.6 w=2.0
x756 786 788 1 0 statinv                *C2
j728 0 788 0 0 enh.1 l=0.8 w=4.0        *Load
```

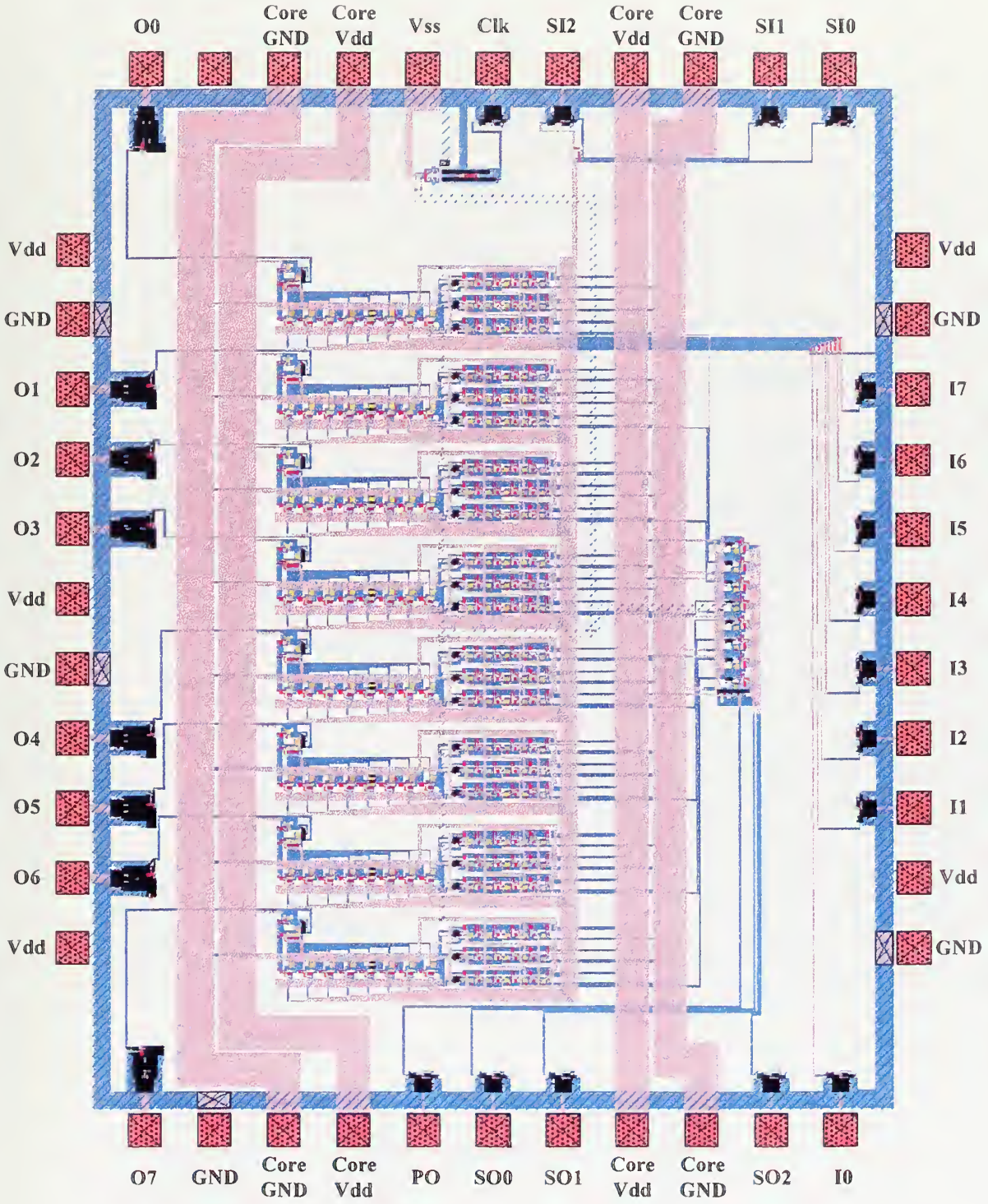
```
*******************************************************************************
*Inputs
*******************************************************************************
x60 6 90 99 0 receiver                  *BI0
j32 0 90 0 0 enh.1 l=0.8 w=4.0          *Load
x61 6 91 99 0 receiver                  *BI1
j33 0 91 0 0 enh.1 l=0.8 w=4.0          *Load
x62 6 92 99 0 receiver                  *BI2
j34 0 92 0 0 enh.1 l=0.8 w=4.0          *Load
x63 6 93 99 0 receiver                  *BI3
j35 0 93 0 0 enh.1 l=0.8 w=4.0          *Load
x64 6 94 99 0 receiver                  *BI4
j36 0 94 0 0 enh.1 l=0.8 w=4.0          *Load
x65 6 95 99 0 receiver                  *BI5
j37 0 95 0 0 enh.1 l=0.8 w=4.0          *Load
x66 6 96 99 0 receiver                  *BI6
j38 0 96 0 0 enh.1 l=0.8 w=4.0          *Load
x67 6 97 99 0 receiver                  *BI7
j39 0 97 0 0 enh.1 l=0.8 w=4.0          *Load

*******************************************************************************
*Multiplexers
*******************************************************************************
*Dynamic 8-to-1 Mux Number 0
x70 90 88 78 68 100 1 2 0 3 4 mednor4
x71 91 88 78 67 101 1 2 0 3 4 mednor4
x72 92 88 77 68 102 1 2 0 3 4 mednor4
x73 93 88 77 67 103 1 2 0 3 4 mednor4
x74 94 87 78 68 104 1 2 0 3 4 mednor4
x75 95 87 78 67 105 1 2 0 3 4 mednor4
x76 96 87 77 68 106 1 2 0 3 4 mednor4
x77 97 87 77 67 107 1 2 0 3 4 mednor4
x80 100 101 102 103 104 105 106 107 108 1 2 0 4 3 smallnor8
x81 108 109 1 2 0 3 4 mininv
j40 109 4 110 2 dp1.1 l=1.6 w=2.0
x82 110 111 1 0 superbuffer
x83 111 112 99 0 driver

*Dynamic 8-to-1 Mux Number 1
x170 90 188 178 168 1100 1 2 0 3 4 mednor4
x171 91 188 178 167 1101 1 2 0 3 4 mednor4
x172 92 188 177 168 1102 1 2 0 3 4 mednor4
x173 93 188 177 167 1103 1 2 0 3 4 mednor4
x174 94 187 178 168 1104 1 2 0 3 4 mednor4
```

```
x175 95 187 178 167 1105 1 2 0 3 4 mednor4
x176 96 187 177 168 1106 1 2 0 3 4 mednor4
x177 97 187 177 167 1107 1 2 0 3 4 mednor4
x180 1100 1101 1102 1103 1104 1105 1106 1107 1108 1 2 0 4 3 smallnor8
x181 1108 1109 1 2 0 3 4 mininv
j140 1109 4 1110 2 dp1.1 l=1.6 w=2.0
x182 1110 1111 1 0 superbuffer
x183 1111 1112 99 0 driver

*Dynamic 8-to-1 Mux Number 2
x270 90 288 278 268 2100 1 2 0 3 4 mednor4
x271 91 288 278 267 2101 1 2 0 3 4 mednor4
x272 92 288 277 268 2102 1 2 0 3 4 mednor4
x273 93 288 277 267 2103 1 2 0 3 4 mednor4
x274 94 287 278 268 2104 1 2 0 3 4 mednor4
x275 95 287 278 267 2105 1 2 0 3 4 mednor4
x276 96 287 277 268 2106 1 2 0 3 4 mednor4
x277 97 287 277 267 2107 1 2 0 3 4 mednor4
x280 2100 2101 2102 2103 2104 2105 2106 2107 2108 1 2 0 4 3 smallnor8
x281 2108 2109 1 2 0 3 4 mininv
j240 2109 4 2110 2 dp1.1 l=1.6 w=2.0
x282 2110 2111 1 0 superbuffer
x283 2111 2112 99 0 driver

*Dynamic 8-to-1 Mux Number 3
x370 90 388 378 368 3100 1 2 0 3 4 mednor4
x371 91 388 378 367 3101 1 2 0 3 4 mednor4
x372 92 388 377 368 3102 1 2 0 3 4 mednor4
x373 93 388 377 367 3103 1 2 0 3 4 mednor4
x374 94 387 378 368 3104 1 2 0 3 4 mednor4
x375 95 387 378 367 3105 1 2 0 3 4 mednor4
x376 96 387 377 368 3106 1 2 0 3 4 mednor4
x377 97 387 377 367 3107 1 2 0 3 4 mednor4
x380 3100 3101 3102 3103 3104 3105 3106 3107 3108 1 2 0 4 3 smallnor8
x381 3108 3109 1 2 0 3 4 mininv
j340 3109 4 3110 2 dp1.1 l=1.6 w=2.0
x382 3110 3111 1 0 superbuffer
x383 3111 3112 99 0 driver

*Dynamic 8-to-1 Mux Number 4
x470 90 488 478 468 4100 1 2 0 3 4 mednor4
x471 91 488 478 467 4101 1 2 0 3 4 mednor4
x472 92 488 477 468 4102 1 2 0 3 4 mednor4
x473 93 488 477 467 4103 1 2 0 3 4 mednor4
```

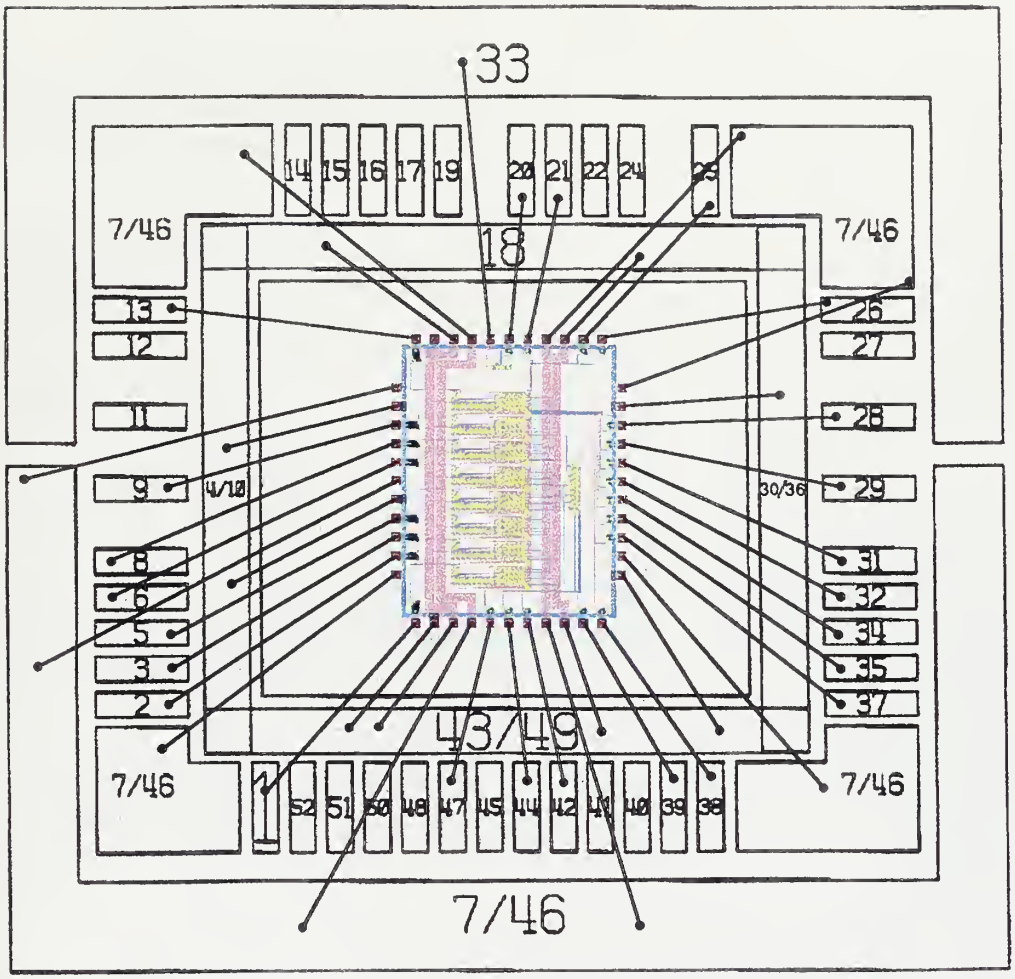
```
x474 94 487 478 468 4104 1 2 0 3 4 mednor4
x475 95 487 478 467 4105 1 2 0 3 4 mednor4
x476 96 487 477 468 4106 1 2 0 3 4 mednor4
x477 97 487 477 467 4107 1 2 0 3 4 mednor4
x480 4100 4101 4102 4103 4104 4105 4106 4107 4108 1 2 0 4 3 smallnor8
x481 4108 4109 1 2 0 3 4 mininv
j440 4109 4 4110 2 dp1.1 l=1.6 w=2.0
x482 4110 4111 1 0 superbuffer
x483 4111 4112 99 0 driver

*Dynamic 8-to-1 Mux Number 5
x570 90 588 578 568 5100 1 2 0 3 4 mednor4
x571 91 588 578 567 5101 1 2 0 3 4 mednor4
x572 92 588 577 568 5102 1 2 0 3 4 mednor4
x573 93 588 577 567 5103 1 2 0 3 4 mednor4
x574 94 587 578 568 5104 1 2 0 3 4 mednor4
x575 95 587 578 567 5105 1 2 0 3 4 mednor4
x576 96 587 577 568 5106 1 2 0 3 4 mednor4
x577 97 587 577 567 5107 1 2 0 3 4 mednor4
x580 5100 5101 5102 5103 5104 5105 5106 5107 5108 1 2 0 4 3 smallnor8
x581 5108 5109 1 2 0 3 4 mininv
j540 5109 4 5110 2 dp1.1 l=1.6 w=2.0
x582 5110 5111 1 0 superbuffer
x583 5111 5112 99 0 driver

*Dynamic 8-to-1 Mux Number 6
x670 90 688 678 668 6100 1 2 0 3 4 mednor4
x671 91 688 678 667 6101 1 2 0 3 4 mednor4
x672 92 688 677 668 6102 1 2 0 3 4 mednor4
x673 93 688 677 667 6103 1 2 0 3 4 mednor4
x674 94 687 678 668 6104 1 2 0 3 4 mednor4
x675 95 687 678 667 6105 1 2 0 3 4 mednor4
x676 96 687 677 668 6106 1 2 0 3 4 mednor4
x677 97 687 677 667 6107 1 2 0 3 4 mednor4
x680 6100 6101 6102 6103 6104 6105 6106 6107 6108 1 2 0 4 3 smallnor8
x681 6108 6109 1 2 0 3 4 mininv
j640 6109 4 6110 2 dp1.1 l=1.6 w=2.0
x682 6110 6111 1 0 superbuffer
x683 6111 6112 99 0 driver
```

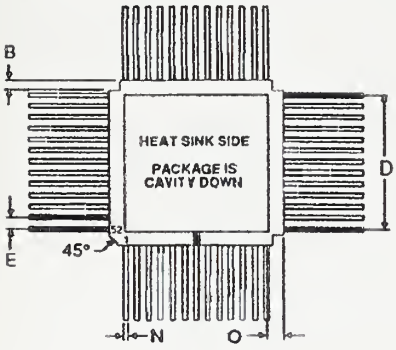
```
*Dynamic 8-to-1 Mux Number 7
x770 90 788 778 768 7100 1 2 0 3 4 mednor4
x771 91 788 778 767 7101 1 2 0 3 4 mednor4
x772 92 788 777 768 7102 1 2 0 3 4 mednor4
x773 93 788 777 767 7103 1 2 0 3 4 mednor4
x774 94 787 778 768 7104 1 2 0 3 4 mednor4
x775 95 787 778 767 7105 1 2 0 3 4 mednor4
x776 96 787 777 768 7106 1 2 0 3 4 mednor4
x777 97 787 777 767 7107 1 2 0 3 4 mednor4
x780 7100 7101 7102 7103 7104 7105 7106 7107 7108 1 2 0 4 3 smallnor8
x781 7108 7109 1 2 0 3 4 mininv
j740 7109 4 7110 2 dp1.1 l=1.6 w=2.0
x782 7110 7111 1 0 superbuffer
x783 7111 7112 99 0 driver

**********************************************************************
* analysis parameters
.options scale=1E-06 post
.trans 10ps 15050ps
.end
```

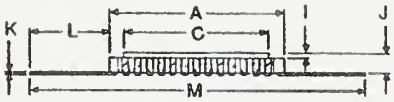
APPENDIX B. TDFL CROSSPOINT SWITCH LAYOUT

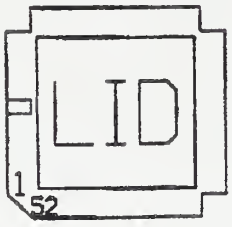


APPENDIX C. TDFL CROSSPOINT SWITCH LAYOUT









Notes: 1) Drawing not to scale.
2) Packages: Ceramic (alumina); Heat sinks: Copper-tungsten; Leads: Alloy 42 with gold plating

Pin	*Function*	*Pin*	*Function*
1-3, 5-6, 8-9, 11-17	Input/Output	4,10,18,30,36,43,49	GND
19-22, 24-29, 31-32	Input/Output	7,46	(-) Supply 1
34-35, 37-42, 44-45	Input/Output	33	Supply 2
47-48, 50-52	Input/Output	23	(substrate)

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Professor Douglas Fouts, Code EC/FS 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Professor Sherif Michael, Code EC/MI 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Lieutenant David Bates 3
   915 Graceland
   Houston, TX 77009

7. Mr. Richard Bates 1
   307 Canadian
   Houston, TX 77009